



TUGAS AKHIR - TF 141581

PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC) PADA PENGENDALIAN KECEPATAN SISTEM MOTOR DC DENGAN KESALAHAN PADA SENSOR

AMALIA ZABDA FASYA
NRP. 02311440000014

Dosen Pembimbing:
Dr. Katherin Indriawati, S.T., M.T.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018

“Halaman ini sengaja dikosongkan”



FINAL PROJECT - TF 141581

**PASSIVE FAULT TOLERANT CONTROL (PFTC)
DESIGN FOR VELOCITY CONTROL OF DC
MOTOR SYSTEM WITH FAULT SENSOR**

**AMALIA ZABDA FASYA
NRP. 02311440000014**

**Supervisor:
Dr. Katherin Indriawati, S.T., M.T.**

**ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018**

“Halaman ini sengaja dikosongkan”

PERNYATAAN BEBAS PLAGIARISME

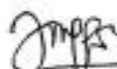
Saya yang bertanda tangan di bawah ini

Nama : Amalia Zabda Fasya
NRP : 023114410000014
Departemen/Prodi : Teknik Fisika/S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "Perancangan *Passive Fault Tolerant Control* (PFTC) pada Pengendalian Kecepatan Sistem Motor DC dengan Kesalahan pada Sensor" adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, 10 Januari 2018
Yang membuat pernyataan,



Amalia Zabda Fasya
NRP. 02311440000014

Halaman ini sengaja dikosongkan

**LEMBAR PENGESAHAN I
TUGAS AKHIR**

**PERANCANGAN *PASSIVE FAULT TOLERANT
CONTROL* (PFTC) PADA PENGENDALIAN
KECEPATAN SISTEM MOTOR DC DENGAN
KESALAHAN PADA SENSOR**

Oleh:

**Amalia Zabda Fasya
NRP. 02311440000014**

Surabaya, 10 Januari 2018

**Menyetujui,
Dosen Pembimbing**

**Dr. Katherin Indriawati, S.T., M.T.
NIPN. 19760523 200012 2 001**

**Mengetahui,
Kepala Departemen
Teknik Fisika FTI-ITS**



**Agus Muhamad Hatta, S.T., M.Si., Ph.D.
NIPN. 19780902 200312 1 002**

“Halaman ini sengaja dikosongkan”

**PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC) PADA PENGENDALIAN
KECEPATAN SISTEM MOTOR DC DENGAN
KESALAHAN PADA SENSOR**




TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh:

AMALIA ZABDA FASYA
NRP. 02311440000014

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Katherin Indriawati, S.T., M.T.  (Pembimbing)
2. Totok Ruki Biyanto, S.T., M.T., Ph.D.  (Penguji I)
3. Hendra Cordova, ST., MT.  (Penguji II)

**SURABAYA
JANUARI, 2018**

“Halaman ini sengaja dikosongkan”

PERANCANGAN *PASSIVE FAULT TOLERANT CONTROL* (PFTC) PADA PENGENDALIAN KECEPATAN SISTEM MOTOR DC DENGAN KESALAHAN PADA SENSOR

Nama : Amalia Zabda Fasya
NRP : 02311440000014
Departemen : Teknik Fisika FTI-ITS
Dosen Pembimbing : Dr. Katherin Indriawati, S.T., M.T.

ABSTRAK

Sistem motor DC pada mobil listrik sering mengalami kegagalan yang disebabkan oleh kesalahan pada komponen sensor. Pada Tugas akhir ini dilakukan perancangan sistem *Passive Fault Tolerant Control* (PFTC) yang dapat mempertahankan kinerja sistem ketika terjadi kesalahan pada komponen sensor. Langkah pertama yang dilakukan adalah merancang sistem pengendali *state feedback* dengan integrator. Langkah kedua adalah merancang *observer* untuk mengestimasi nilai variabel keadaan yaitu kecepatan dan arus. Dalam hal ini *gain observer* dicari menggunakan metode optimasi berbasis LMI (*Linier Matrix Inequalities*). Sistem PFTC bekerja dengan menggunakan hasil estimasi keadaan dari *observer*, menggantikan hasil pengukuran. Dari hasil pengujian didapatkan nilai ketetapan kesalahan yang terbaik adalah 0,3. Kesalahan bias maksimum yang mampu diatasi adalah 60%. Respon sistem dengan PFTC tidak menghasilkan *maximum undershoot* sedangkan sistem tanpa PFTC menghasilkan *maximum undershoot* sebesar 63%. Sistem dengan PFTC tidak menghasilkan *error steady state* sedangkan sistem tanpa PFTC menghasilkan *error* sebesar 58%. Kesalahan sensitifitas maksimum yang mampu diatasi adalah 90%. Respon sistem dengan PFTC tidak menghasilkan *error steady state* sedangkan sistem tanpa PFTC menghasilkan *error* 50.7%. Selain itu PFTC mampu mengatasi ketidakpastian sebesar 10% pada tahanan (R).

Kata Kunci : Sensor, Kontrol *State Feedback Integrator*, Sistem PFTC, Sistem Motor DC.

“Halaman ini sengaja dikosongkan”

**PASSIVE FAULT TOLERANT CONTROL (PFTC)
DESIGN FOR VELOCITY CONTROL OF DC MOTOR
SYSTEM WITH FAULT SENSOR**

Name : Amalia Zabda Fasya
NRP : 02311440000014
Department : Teknik Fisika FTI-ITS
Supervisors : Dr. Katherin Indriawati, S.T., M.T.

ABSTRACT

DC motor systems in electric cars often experience failure caused by errors in the sensor components. This final project designed a system Passive Fault Tolerant Control (PFTC) which can maintain the performance of the system when an error occurs on the sensor component. The first step was designing a state feedback control system with an integrator. The second step was designing an observer to estimate the value of state variables which is speed and current. In this case the gain observer searched by using the optimization method based on LMI (Linear Matrix Inequalities). The PFTC system works by using the state estimation results from the observer, replacing the measurement results. From the test results obtained the best failure constant value of 0.3. The maximum bias error that can be overcome is 60%. Response system with PFTC did not generate maximum undershoot while system without PFTC produced maximum undershoot equal to 63%. System with PFTC did not generate steady state error while system without PFTC generated error equal to 58%. The maximum sensitivity error that can be overcome is 90%. Response system with PFTC did not generate steady state error while system without PFTC generated error of 50.7%. In addition PFTC is able to overcome the uncertainty of 10% in coil resistance (R).

Keywords: *Sensor, State Feedback Control With Integrator, PFTC System, DC Motor System.*

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya serta shalawat dan salam kepada Nabi Muhammad SAW sehingga dapat terselesaikannya tugas akhir beserta laporan tugas akhir yang berjudul **“Perancangan *Passive Fault Tolerant Control* (PFTC) Pada Pengendalian Kecepatan Sistem Motor DC Dengan Kesalahan Pada Sensor”**.

Penulis telah banyak memperoleh bantuan dari berbagai pihak dalam penyelesaian tugas akhir dan laporan tugas akhir ini. Penulis mengucapkan terima kasih kepada:

1. Bapak Agus Muhamad Hatta, S.T., M.Si., Ph.D selaku Ketua Departemen Teknik Fisika ITS yang telah memberikan ilmu, bimbingan serta sarana dan prasarana selama menempuh pendidikan di Teknik Fisika ITS.
2. Ibu Dr. Katherin Indriawati, S.T., M.T. selaku dosen pembimbing yang telah dengan sabar memberikan petunjuk, ilmu, serta bimbingan dalam pengerjaan tugas akhir ini.
3. Bapak Totok Ruki Biyanto, S.T., M.T., Ph.D selaku Kepala Laboratorium Rekayasa Instrumensi yang telah memberikan ilmu, petunjuk, dan kemudahan perizinan.
4. Bapak Ir. Zulkifli, Msc selaku dosen wali yang telah membimbing penulis selama perkuliahan dan memberikan ilmu serta nasihat yang sangat bermanfaat.
5. Seluruh dosen Teknik Fisika ITS atas segala ilmu, bimbingan, petunjuk, dan nasihat yang sangat bermanfaat.
6. Keluarga, atas segala cinta, kasih sayang, doa, perhatian, serta dukungan moril dan materiil yang telah diberikan.
7. Seluruh teman seperjuangan dalam tugas akhir, yaitu terima kasih untuk semuanya.
8. Seluruh teman-teman Departemen Teknik Fisika ITS angkatan 2014, terima kasih untuk semuanya.
9. Seluruh karyawan dan civitas akademika Teknik Fisika ITS, terima kasih atas segala bantuan dan kerjasamanya.

10.Semua pihak yang tidak dapat disebutkan satu persatu, terimakasih atas bantuannya.

Penulis sadar bahwa penulisan laporan tugas akhir ini masih terdapat kekurangan. Namun, semoga laporan ini dapat memberikan kontribusi yang berarti dan menambah wawasan yang bermanfaat bagi pembaca, keluarga besar Teknik Fisika ITS khususnya, dan civitas akademika ITS pada umumnya. Semoga laporan tugas akhir ini dapat bermanfaat sebagai referensi pengerjaan laporan tugas akhir bagi mahasiswa lainnya.

Surabaya, 10 Januari 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
COVER PAGE.....	iii
PERNYATAAN BEBAS PLAGIARISME.....	v
LEMBAR PENGESAHAN I	vii
LEMBAR PENGESAHAN II	ix
ABSTRAK	xi
<i>ABSTRACT</i>	xiii
KATA PENGANTAR.....	xv
DAFTAR ISI.....	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR NOTASI	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Sistematika Laporan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Motor <i>Direct Current</i> (DC) Magnet Permanen.....	5
2.2 Pemodelan Motor DC	8
2.3 Sistem Kontrol <i>State Feedback</i> dengan Integrator.....	131
2.4 Konsep Kestabilan dan Performansi Robust.....	21
2.5 Sistem <i>Passive Fault Tolerant Control</i> (PFTC)....	22
2.6 Optimasi dengan LMI.....	24
BAB III METODOLOGI PENELITIAN.....	27
3.1 Pemodelan Sistem Motor DC.....	30
3.2 Perancangan Sistem Kontrol <i>State Feedback</i>	31
3.3 Perancangan <i>Observer</i>	33
3.4 Penentuan Gain <i>Observer</i>	39
3.5 Uji Performansi PFTC	43
BAB IV ANALISIS DATA DAN PEMBAHASAN	45
4.1 Uji <i>Tracking</i> Kontroler.....	45
4.2 Uji Hasil Estimasi Observer.....	45

4.3 Uji Performansi Kesalahan Sensor.....	47
BAB V KESIMPULAN DAN SARAN	63
5.1 Kesimpulan	63
5.2 Saran.....	64
DAFTAR PUSTAKA.....	65
LAMPIRAN	69

DAFTAR GAMBAR

Gambar 2. 1	Motor DC	6
Gambar 2. 2	<i>Tachometer</i>	6
Gambar 2. 3	Rangkaian motor DC	9
Gambar 2. 4	Diagram blok sistem pengendalian tertutup	11
Gambar 2. 5	Diagram blok kontrol <i>state feedback</i> dengan <i>integrator</i>	12
Gambar 2. 6	Diagram sistem kontrol secara umum.....	13
Gambar 2. 7	Diagram kontrol <i>robust</i>	14
Gambar 2. 8	Konfigurasi gangguan dalam bentuk penambahan (<i>additive</i>)	15
Gambar 2. 9	Konfigurasi gangguan dalam bentuk perkalian input.....	15
Gambar 2. 10	Konfigurasi gangguan dalam bentuk perkalian output.....	16
Gambar 2. 11	Sistem kontrol robust secara umum	16
Gambar 2. 12	Ilustri grafik luasan H_2 -norm.....	20
Gambar 2. 13	Ilustri grafik luasan H_∞ -norm.....	20
Gambar 3. 1	Diagram alir peneluitian tugas akhir.....	28
Gambar 3. 2	Diagram blok sistem pengendalian kecepatan motor	31
Gambar 4.1	Respon uji kontroler.....	45
Gambar 4.2	Hasil perbandingan kesalahan aktual dengan estimasi kesalahan sensor tanpa kesalahan	46
Gambar 4.3	Hasil perbandingan kesalahan aktual dengan estimasi kesalahan sensor dengan kesalahan.....	46
Gambar 4. 4	Respon dengan $F_s = 0.3$ kesalahan bias 13%.....	47
Gambar 4. 5	Respon dengan $F_s = 0.3$ kesalahan bias 47%.....	49
Gambar 4. 6	Respon dengan $F_s = 0.3$ kesalahan bias 60%.....	50
Gambar 4. 7	Respon dengan $F_s = 0.3$ kesalahan bias 100%.....	51
Gambar 4. 8	Respon dengan $F_s = 0.3$ kesalahan sensitifitas 10%	54
Gambar 4. 9	Respon dengan $F_s = 0.3$ kesalahan sensitifitas 50%	55
Gambar 4. 10	Respon dengan $F_s = 0.3$ kesalahan sensitifitas 90%	56

Gambar 4.11 Respon dengan $F_s = 0.3$ kesalahan sensitifitas
100%57

Gambar 4. 12 Respon dengan ketidakpastian 10%60

DAFTAR TABEL

Tabel 4. 1	Performansi sistem $F_s = 0.3$ kesalahan bias 13%	48
Tabel 4. 2	Performansi sistem $F_s = 0.3$ kesalahan bias 47%	49
Tabel 4. 3	Performansi sistem $F_s = 0.3$ kesalahan bias 60%	50
Tabel 4. 4	Performansi sistem $F_s = 0.3$ kesalahan bias 100% ...	51
Tabel 4. 5	Perbandingan Hasil Respon Nilai F_s	53
Tabel 4. 6	Performansi sistem $F_s = 0.3$ kesalahan sensitifitas 10%	54
Tabel 4. 7	Performansi sistem $F_s = 0.3$ kesalahan sensitifitas 50%	55
Tabel 4. 8	Performansi sistem $F_s = 0.3$ kesalahan sensitifitas 90%	56
Tabel 4. 9	Performansi sistem $F_s = 0.3$ kesalahan sensitifitas 100%	57
Tabel 4. 10	Perbandingan Hasil Respon Nilai F_s	59
Tabel 4. 11	Performansi dengan ketidakpastian 10%	60

“Halaman ini sengaja dikosongkan”

DAFTAR NOTASI

e_a	= tegangan yang dikenakan pada jangkar, volt
e_b	= ggl - gaya gerak listrik balik, volt
I	= arus kumparan jangkar, ampere
R	= Tahanan kumparan jangkar, ohm
L	= induktansi kumparan jangkar, henry
V	= tegangan masukan servo, volt
b	= Koefisien gesekan viskos ekivalen dari motor dan beban pada poros motor, N.m/rad.dt
K	= Konstanta torsi motor
R	= Tahanan kumparan jangkar, ohm
J	= momen inersia ekuivalen dari motor, kg.m ²
L	= induktansi kumparan jangkar, henry
K_p	= konstanta/ <i>gain</i> pengendali <i>proportional</i>
e	= <i>error</i> atau sinyal kesalahan = <i>setpoint</i> – keluaran sistem
$T_{zw}(s)$	= fungsi alih
A	= Matriks keadaan (<i>state</i>)
B	= Matriks masukan/matriks kendali
C	= Matriks keluaran
D	= Matriks gangguan
x	= vector keadaan
$u(t)$	= Masukan kendali
$y(t)$	= Vektor keluaran
$x(t)$	= Vektor keadaan (<i>state</i>)
$s_f(t)$	= Sinyal actual
F_s	= Matrik kesalahan sensor
f_s	= Vektor kesalahan sensor
F_a	= Matrik kesalahan aktuator
f_a	= Vektor kesalahan aktuator
p	= <i>poles</i>
\dot{x}	= Turunan x
\hat{x}	= Hasil estimasi dari x
$\hat{\dot{x}}$	= Turunan hasil estimasi x
y	= Vektor output pengukuran

“Halaman ini sengaja dikosongkan”

BAB I

PENDAHULUAN

1.1 Latar Belakang

Motor arus searah atau motor *Direct Current* (DC) merupakan sebuah perangkat elektromagnetis yang berfungsi mengubah energi listrik menjadi energi mekanik. Motor ini paling banyak digunakan dalam dunia industri dan dunia otomotif untuk sumber penggerak dari mobil listrik (Koksal, 2007).

Sistem pengendalian dalam industri banyak menggunakan pengendali konvensional. Pengendali konvensional untuk kecepatan motor DC memiliki kinerja yang buruk ketika terjadi perubahan beban atau dinamika motor (G.Rigatos, 2007). Hal ini dikarenakan performa sistem rentan terhadap kesalahan pada sensor dan aktuator. Sensor dan aktuator merupakan komponen yang mudah sekali rusak karena langsung berinteraksi dengan lingkungan di lapangan. Kesalahan pada sensor motor DC mengakibatkan kinerja dari sistem menurun dan juga berdampak pada performansi sistem *tracking* cenderung menurun. Kesalahan pada komponen tersebut dapat diatasi dengan sistem pengendali yang mampu mentolelir kesalahan secara otomatis sehingga kegagalan dari suatu sistem dapat diatasi.

Sistem kontrol yang mampu mengatasi terjadinya kegagalan komponen secara otomatis dengan menjaga stabilitas sistem dan kinerja yang dapat diterima dalam suatu kegagalan disebut dengan *Fault Tolerant Control System* (FTCS) (Indriawati, 2015). *Fault Tolerant Control System* (FTCS) dapat dibagi menjadi dua tipe yaitu *Passive Fault Tolerant Control* (PFTC) dan *Active Fault Tolerant Control* (AFTC) (Zhang & Jiang, 2008). Pada *Passive Fault Tolerant Control* (PFTC), didesain untuk menjadi sistem pengendali yang *robust* dari kesalahan suatu komponen. Teori *robust* memiliki karakteristik yang tidak peka terhadap perubahan parameter, kesalahan model dan gangguan (Tsui C. , 2004). Awalnya teori *robust* hanya untuk mengatasi ketidakpastian (*uncertainty*). Perkembangan selanjutnya teori *robust* juga

digunakan untuk mengatasi kesalahan pada komponen sensor dan aktuator.

Dalam penelitian sebelumnya telah dilakukan perancangan sistem *Passive Fault Tolerant Control (PFTC)* pada sistem servo dengan kesalahan sensor (Noboru Sebe, 2013). Dari perancangan sistem PFTC tersebut tidak menggunakan *observer* sehingga besarnya kesalahan tidak bisa diestimasi yang menyebabkan performa sistem menurun. Oleh karena itu diperlukan pengembangan untuk memperbaiki metode tersebut dengan menggunakan metode terbaru yaitu merancang sistem PFTC dengan menggunakan *observer* sehingga dapat mengestimasi besar kesalahan yang terjadi pada sistem agar performansi sistem terjaga. Pada tugas akhir ini dilakukan perancangan sistem *Passive Fault Tolerant Control (PFTC)* pada pengendalian kecepatan sistem motor DC dengan kesalahan pada sensor. Dengan menggunakan sistem PFTC dapat merancang suatu sistem pengendali yang *robust* dari kesalahan komponen, sehingga dapat terjaga kestabilan dan kinerja sistem.

1.2 Rumusan Masalah

Dari latar belakang yang telah diuraikan diatas, maka didapatkan permasalahan dalam tugas akhir ini sebagai berikut:

- a. Bagaimana mendesain *observer* yang dapat mengestimasi kesalahan dan menghasilkan *state* yang benar ?
- b. Bagaimana merancang *Passive Fault Tolerant Control (PFTC)* pada pengendalian kecepatan sistem motor DC dengan menggunakan *observer* ?

1.3 Tujuan

Tujuan dari penelitian tugas akhir ini adalah merancang sistem pengendalian pada sistem motor DC dengan menggunakan sistem pengendali *robust* atau *Passive Fault Tolerant Control (PFTC)* yang mampu mengatasi kesalahan pada sensor sehingga performansi sistem tetap terjaga.

1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut.

- a. Komponen sensor yang digunakan adalah *tachometer* dengan kesalahan berupa adanya *zero* bias atau perubahan nilai *zero* kecepatan sensor yang menyebabkan kesalahan bias dan sensitifitas.
- b. Variabel yang dikontrol adalah kecepatan motor DC.
- c. Variabel yang dimanipulasi adalah tegangan motor DC melalui sensor *tachometer*.
- d. Motor DC yang digunakan tipe magnet permanen.
- e. Ketidakpastian muncul di tahanan kumparan jangkar (R) dengan satuan *ohm*.
- f. Kesalahan yang terjadi adalah kesalahan minor berupa kesalahan bias dan kesalahan sensitifitas.

1.5 Sistematika Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

- a. BAB I PENDAHULUAN
Pada bab I ini terdiri dari latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika laporan.
- b. BAB II TEORI PENUNJANG
Pada bab II ini dibahas mengenai teori-teori yang berkaitan dengan penelitian yang akan dilakukan, seperti teori sistem servo motor DC, pemodelan kecepatan motor DC, kontrol *feedback* kontrol *robust*, sistem PFTC, dan *observer*.
- c. BAB III METODOLOGI PENELITIAN
Pada bab III ini berisi mengenai rancangan dari penelitian yang dilakukan, metode, dan langkah-langkah dalam penelitian.
- d. BAB IV ANALISIS DATA DAN PEMBAHASAN
Pada bab IV ini berisi tentang analisis hasil perancangan sistem *passive fault tolerant control*.
- e. BAB V KESIMPULAN DAN SARAN
Pada bab V ini diberikan kesimpulan tentang tugas akhir yang telah dilakukan berdasarkan data-data yang diperoleh serta

diberikan saran sebagai penunjang maupun pengembangan untuk tugas akhir yang selanjutnya

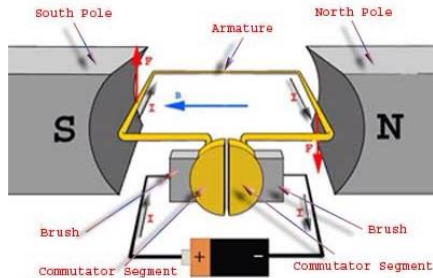
BAB II

TINJAUAN PUSTAKA

2.1 Motor *Direct Current* (DC) Magnet Permanen

Motor *Direct Current* (DC) adalah mesin listrik yang prinsip kerjanya mengubah energi listrik arus searah menjadi energi mekanik. Energi mekanik yang dihasilkan berupa energi gerak putar pada poros motor. Konversi energi tersebut dilakukan dalam waktu yang cepat dan merupakan implementasi hukum Lorentz, yaitu hukum yang berlaku untuk aliran listrik pada kawat penghantar yang berada di dalam medan magnet (Koksal, 2007).

Motor DC magnet permanen seperti ditunjukkan pada Gambar 2.1 tersusun dari beberapa bagian, yaitu magnet permanen, kumparan jangkar (*armature*), komutator (*commutator*), dan sikat (*brush*). Sumber tegangan DC yang digambarkan dengan baterai dimana terhubung dengan sikat karbon pada masing-masing kutub baterai sehingga dapat tercipta arus listrik (I). Pada motor DC arus mengalir dari kutub positif ke kutub negatif melewati sikat karbon, kemudian menuju komutator, dilanjutkan kumparan jangkar (*armature*), kembali ke komutator, sikat karbon, dan ke kutub negatif baterai. Pada bagian stator motor, kumparan medan stator dapat menghasilkan fluks dari kutub utara ke kutub selatan. Didalam kumparan magnet terdapat arus pada kumparan jangkar yang mengandung fluks sehingga menghasilkan suatu gaya, yang disebut gaya Lorentz (F). Komponen komutator yang selalu ikut berputar dengan rotor dan sikat karbon yang selalu diam sebagai komponen yang akan menjaga arah arus listrik untuk selalu tetap, yaitu mengalir dari sisi kiri kawat menuju ke sisi kanan. Selama masih ada arus listrik yang mengalir ke kumparan jangkar, arah arus listrik yang selalu tetap di setiap setengah putaran rotor ini yang akan membuat rotor motor listrik selalu berputar.



Gambar 2. 1 Motor DC (Burrige & Qu, 2003)

Secara umum, pengukuran motor DC berhubungan dengan arus stator atau rotor dan tegangan (pengukuran elektrik) dan perpindahan poros sudut atau kecepatan sudut (pengukuran mekanik) (Delgado, 2007). Pengukuran pada motor DC dengan menggunakan sensor untuk menghitung kecepatan dan arus. Sensor yang digunakan untuk mengukur kecepatan dengan menggunakan *tachometer* dan digital *encoder* sedangkan untuk pengukuran arus menggunakan *Hall-effect* atau sensor arus *shuntresistor*. Aktuator yang digunakan pada motor DC yaitu menggunakan *servoamplifier*.

a. Tachometer

Sensor kecepatan yang umum digunakan pada sistem pengendalian motor dengan menggunakan *Tachometer* DC, yang merupakan suatu perangkat elektronik yang mengubah energi mekanik ke energi listrik. Tachometer ini biasanya menampilkan *revolutions per-minute* (RPM) pada sebuah pengukur skala analog (Kuo, 1995).



Gambar 2. 2 Tachometer (Kuo, 1995)

Kelebihan dari *tachometer* yaitu untuk menjaga inersia turun yang dapat diatasi dengan penggunaan sikat, sedangkan

kelemahannya adalah penggunaan sikat untuk menjaga inersia dapat aus. Keluaran dari rangkaian tachometer berupa sinyal frekuensi diubah menjadi tegangan oleh rangkaian pengubah frekuensi menjadi tegangan. Jika respon dinamik sensor jauh lebih cepat dibandingkan dengan respon proses, maka konstanta waktu (*time constan*) dan *death time* pada sensor dapat diabaikan, sehingga fungsi alih sensor dapat didekati dengan penguatan (gain saja). Sehingga gain dari *tachometer* adalah

$$E = K \omega_m \quad (2.1)$$

Dimana

E = tegangan keluaran dari *tachometer* (volt)

K = gain

ω_m = kecepatan putar motor(rpm)

Kesalahan pada sensor *tachometer* dapat berupa adanya zero bias atau perubahan nilai zero kecepatan pada sensor, kesalahan pembacaan akibat perubahan sensitifitas dan kalibrasi yang buruk. Kesalahan tersebut menyebabkan kesalahan berupa kesalahan minor pada sensor. Kesalahan sensor dibedakan menurut tingkat keparahannya (*severity*), yaitu kesalahan minor dan kesalahan mayor (Noura, Theilliol, Ponsart, & Chamseddine, 2009). Dalam hal ini, tingkat kerusakan diukur dari dampak yang ditimbulkan terhadap performa sistem. Kesalahan minor adalah kesalahan dengan tingkat keparahan yang kecil sehingga sistem masih dapat bekerja dengan performa yang diijinkan. Pada sensor, kesalahan pembacaan akibat perubahan sensitivitas dan adanya *bias* merupakan bentuk kesalahan minor. Sedangkan kesalahan mayor adalah kesalahan yang menyebabkan sistem tidak mungkin dapat mempertahankan performa yang ditetapkan. Kesalahan bias dan sensitifitas dapat dituliskan dalam bentuk persamaan berikut :

$$\begin{aligned} y &= y_o + \beta \\ y &= y_o \alpha \end{aligned} \quad (2.2)$$

Dimana:

y = sinyal kontrol *input residual* (sinyal kontrol *input* yang masuk pada sensor setelah diberi pembebanan berupa *gain* kesalahan)

y_o = sinyal kontrol *input* (sinyal kontrol *input* sebelum diberi pembebanan berupa *gain* kesalahan)

α = *gain* kesalahan *sensitifitas*

β = *gain* kesalahan bias

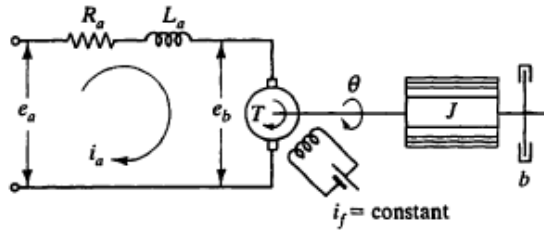
b. Servo amplifier

Merupakan actuator yang digunakan untuk menggerakkan motor DC. *Servo amplifier* adalah sebuah penguat yang mengolah sinyal error untuk mengoreksi perbedaan antara sinyal input (referensi) dengan sinyal umpan balik (*feedback*) sehingga menghasilkan output yang diharapkan. Servo amplifier terdiri dari comparator yang mengolah sinyal error dan power amplifier yang menguatkan sinyal agar mampu menggerakkan servomotor (Setyaningrum, 2012).

2.2 Pemodelan Motor DC

Suatu sistem disusun dari bermacam-macam komponen yang saling berhubungan dalam menjalankan suatu aksi untuk mencapai tujuan yang diinginkan. Motor DC adalah termasuk dalam sistem komponen elektro-mekanik.

Suatu prasyarat dasar pada kebanyakan strategi yang dipergunakan dalam membangun suatu sistem kendali adalah kemampuan dalam memodelkan secara matematika dari sistem yang akan dikendalikan. Model matematika dari suatu sistem dapat berupa persamaan diferensial, fungsi alih atau persamaan ruang keadaan (Ogata, 1997).



Gambar 2. 3 Rangkaian Motor DC (Wai, 2007)

Berdasarkan Gambar 2.3, diasumsikan bahwa arus medan (i_f) dan nilai fluks konstan yang menyebabkan torsi memiliki arah sesuai arus kumparan jangkar, sehingga dapat ditulis dalam bentuk persamaan :

$$T = K_t \cdot i \quad (2.3)$$

Dimana K_t adalah konstanta torsi dan i adalah arus kumparan jangkar. Sedangkan, ketika gaya dibangkitkan akan timbul gaya gerak listrik e_b yang berbanding lurus dengan kecepatan rotasi motor ω , sehingga

$$e_b = K_b \cdot \omega \quad (2.4)$$

dimana e_b adalah emf balik dan K_b adalah konstanta emf balik. Dalam unit SI, konstanta torsi motor dan gaya gerak listrik emf balik adalah sama, sehingga $K_t = K_b$; dengan demikian dapat digunakan konstanta K untuk menyatakan keduanya.

Untuk bagian mekanik, momen inersia rotor (J) dan gesekan viskositas (B), menghasilkan torsi dengan persamaan sebagai berikut

$$J \frac{d\omega}{dt} + b\omega = T = K \cdot i \quad (2.5)$$

Kecepatan kumparan jangkar motor DC dikendalikan oleh tegangan kumparan jangkar e_a dengan rangkaian kumparan jangkar

berupa hambatan jangkar (R) dan induktansi jangkar (L) sehingga persamaan untuk rangkaian kumparan jangkar adalah :

$$L \frac{di}{dt} + Ri + e_b = e_a \quad (2.6)$$

Jika persamaan (2.4) dimasukan ke dalam persamaan (2.6) sehingga

$$L \frac{di}{dt} + Ri = V - K\omega \quad (2.7)$$

Dengan menggunakan transformasi *Laplace* maka persamaan (2.6) dan (2.7) dapat diubah menjadi persamaan dalam bentuk domain s .

$$s(Js + b)\omega(s) = KI(s) \quad (2.8)$$

$$(Ls + R)I(s) = V(s) - K\omega(s) \quad (2.9)$$

Dari persamaan (2.8) $I(s)$ dapat dituliskan

$$I(s) = \frac{V(s) - K\omega(s)}{R + Ls} \quad (2.10)$$

Kemudian disubstitusikan ke persamaan (2.5), menjadi

$$Js\omega(s) + b\omega(s) = K \frac{(V(s) - K\omega(s))}{R + Ls} \quad (2.11)$$

Fungsi transfer masukan tegangan berbanding dengan kecepatan motor adalah

$$\frac{\omega(s)}{V(s)} = \frac{K}{(R + Ls)(Js + b) + KKb} \quad (2.12)$$

L sering kali diabaikan karena nilainya sangat kecil dibandingkan dengan R , sehingga

$$\frac{\omega(s)}{V(s)} = \frac{K}{RJs + Rb + K Kb} \quad (2.13)$$

Dalam bentuk persamaan ruang keadaan, persamaan (2.5) dan (2.7) dapat dituliskan dalam bentuk *state space* dengan memilih kecepatan rotasi dan arus motor sebagai variabel *state* dan tegangan sebagai masukan serta kecepatan rotasi sebagai keluaran:

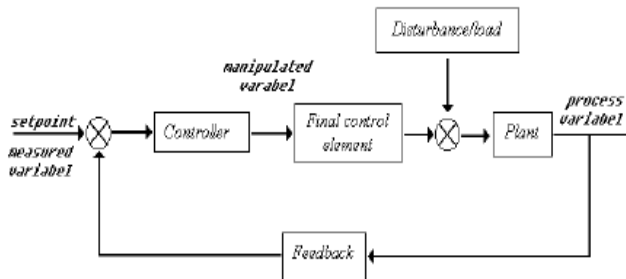
$$\frac{d}{dt} \begin{bmatrix} \omega \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V \quad (2.14)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} \quad (2.15)$$

2.3 Sistem Kontrol *State Feedback* dengan *Integrator*

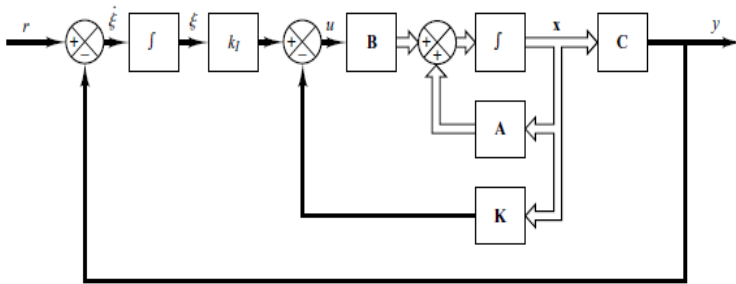
Sistem kontrol bekerja dengan membandingkan nilai sebenarnya dari keluaran sistem secara keseluruhan dengan mengacu pada sinyal acuan (nilai yang dikehendaki), menentukan penyimpangan, menghasilkan sinyal kontrol yang akan mengurangi penyimpangan menjadi nol atau nilai yang kecil.

Sinyal keluaran sistem yang dibandingkan dengan sinyal acuan itulah yang disebut dengan sinyal umpan balik. Oleh sebab itu sistem kontrolnya dinamakan sistem kontrol umpan balik. Sedangkan upaya untuk membuat kesalahan sekecil mungkin tersebut dinamakan aksi kontrol.



Gambar 2. 4 Diagram Blok Sistem Pengendalian Tertutup
(Basilio, 2002)

Kontrol *feedback* dapat berupa kontrol *feedback* dengan *integrator* seperti yang ditunjukkan oleh Gambar 2.8 . Jika dalam suatu sistem ditambahkan dengan *integrator* maka dapat menghilangkan *error steady state*. Jika *integrator* diletakkan sebelum gangguan, maka ini juga dapat membatalkan gangguan masuk ke dalam kondisi yang sudah *steady* (Ogata, 1997). Oleh sebab itu diagram blok sistem pengendalian tertutup dapat diubah menjadi bentuk seperti berikut :



Gambar 2. 5 Diagram Blok Kontrol State Feedback dengan Integrator (Ogata, 1997)

Persamaan untuk kontrol *state feedback* dengan *integrator* dapat dituliskan sebagai berikut :

$$u = K \cdot x + K_i \cdot \int e \cdot dt \quad (2.16)$$

Dimana $K_c = [K \quad K_i]$

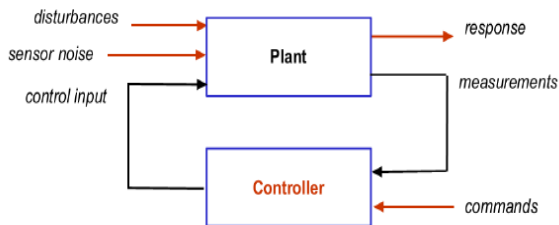
Beberapa karakteristik sistem dalam algoritma pengendali sistem *closed loop* yang juga harus diketahui, yaitu *rise time*, % *overshoot*, *settling time*, *time constant*, dan *error steady state* (Ogata, 1997).

1. *Rise time* adalah waktu yang dibutuhkan dari *output* sistem melebihi 90% dari tingkat saat suatu sistem dijalankan.
2. % *overshoot* adalah *peak level* lebih tinggi dari keadaan *steady state*.

3. *Settling time* adalah waktu yang dibutuhkan sistem hingga stabil. Atau ukuran waktu yang menyatakan respon telah masuk $\pm 5\%$ atau $\pm 2\%$ atau $\pm 0,5\%$ dari respon *steady state*.
4. *Time constant* adalah ukuran waktu yang menyatakan kecepatan respon, yang diukur mulai $t = 0$ sampai dengan respon mencapai 63,2% dari respon *steady state*.
5. *Error steady state* adalah nilai *error* yang dihasilkan dari keadaan *steady* ketika belum mencapai *set point* yang diinginkan.

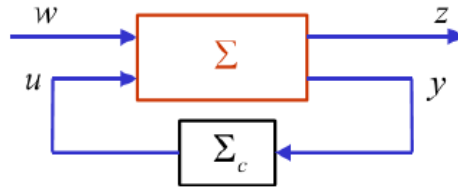
2.4 Konsep Kestabilan dan Performansi *Robust*

Sistem pengendalian bertujuan untuk menghasilkan respon keluaran sistem sesuai dengan referensi yang diinginkan. Secara umum blok diagram dari sistem kontrol sebagai berikut:



Gambar 2. 6 Diagram sistem kontrol secara umum

Suatu sistem dikatakan memiliki sifat *robust* jika sistem tersebut memiliki kemampuan bertahan terhadap keadaan disekitar. Jadi, suatu sistem kontrol yang *robust* adalah sistem kontrol yang mampu bekerja dengan baik (mempertahankan kestabilan), walaupun terdapat gangguan dalam pemodelan yang tidak diketahui karakteristiknya (Kemin Zhou, 1996). Desain dan diagram masalah kontrol *robust* secara umum sebagai berikut :



Gambar 2. 7 Diagram Kontrol *Robust*

Dari Gambar 2.7 diketahui bahwa komponen w merupakan *exogenous input* berupa gangguan, sensor noise, dan sinyal referensi. Komponen z merupakan semua sinyal yang ingin kita kontrol berupa *tracking error* antara sinyal referensi dan output dari plant. Komponen y merupakan output dari sensor yang akan masuk ke controller. Sedangkan untuk komponen u berisi semua kontrol input yang masuk ke plant (John Doyle, 1990).

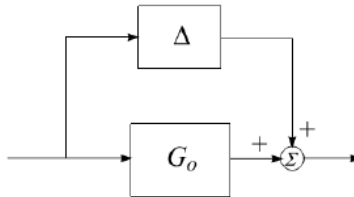
2.4.1 Ketidakpastian (*Uncertainty*)

Sistem fisik secara khusus mengalami gangguan yang bervariasi yang menyebabkan adanya ketidakpastian (*uncertainty*) pada model matematik dari suatu sistem. Ketidakpastian dibagi menjadi dua yaitu tak-terstruktur (*unstructured*) apabila bentuk yang diketahui hanya batas atas (*upper*) dan batas bawah (*lower*) dan dikatakan terstruktur (*structured*) apabila diketahui model rinci. Contoh ketidakpastian terstruktur adalah variasi parametric dalam dinamika *plant* seperti variasi *pole* dan *zero* fungsi transfer *plant*. Sedangkan contoh untuk ketidakpastian tak terstruktur meliputi ketidakpastian yang bergantung pada frekuensi, seperti mode frekuensi tinggi yang biasanya kita abaikan dalam peodelan dinamika *plant* (Oliveira, 2006).

Gangguan dinamik yang terjadi dalam sistem dapat dibangun menjadi suatu blok gangguan (Δ). Dengan sistem actual (sistem dengan gangguan) / $G_p(s)$ dan model nominal plant / $G_0(s)$ sehingga dapat dicari fungsi alih gangguan yang terjadi dalam suatu sistem. Fungsi alih gangguan dapat dibagi menjadi tiga macam yaitu (Kemin Zhou, 1996) :

- Gangguan dalam bentuk penambahan (*additive*)

Diagram blok dari gangguan dalam bentuk penambahan ini adalah sebagai berikut:

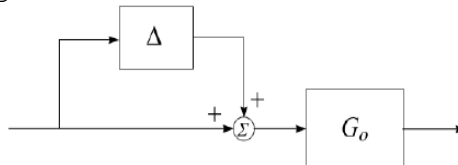


Gambar 2. 8 Konfigurasi gangguan dalam bentuk penambahan (*additive*)

Fungsi alih dalam bentuk penambahan dapat dimodelkan sebagai berikut :

$$G_P(s) = G_0(s) + \Delta(s) \quad (2.17)$$

- Gangguan dalam bentuk perkalian input (*input multiplicative*)
Diagram blok dari gangguan dalam bentuk perkalian input ini adalah sebagai berikut:

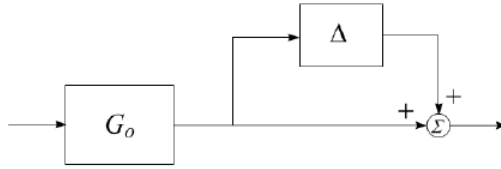


Gambar 2. 9 Konfigurasi gangguan dalam bentuk perkalian input

Fungsi alih dalam bentuk perkalian input dapat dimodelkan sebagai berikut :

$$G_P(s) = G_0(s)[I + \Delta(s)] \quad (2.18)$$

- Gangguan dalam bentuk perkalian output (*output multiplicative*)
Diagram blok dari gangguan dalam bentuk perkalian output ini adalah sebagai berikut:



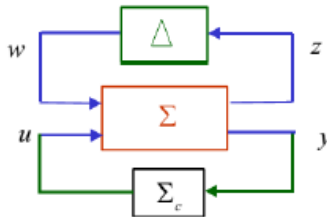
Gambar 2. 10 Konfigurasi gangguan dalam bentuk perkalian output

Fungsi alih dalam bentuk perkalian output dapat dimodelkan sebagai berikut :

$$G_p(s) = G_o(s)[I + \Delta(s)] \quad (2.19)$$

2.4.2 Kestabilan *Robust*

Stabilitas suatu system mencakup persoalan gangguan dan ketidakpastian. Dalam memodelkan sistem secara umum dapat dimisalkan dengan system yang dihubungkan dalam struktur tiga blok seperti Gambar 2.11 yang mengandung blok ketidakpastian Δ (John Doyle, 1990).



Gambar 2. 11 Sistem Kontrol Robust Secara Umum

Kestabilan *Robust* dapat diselidiki di domain frekuensi dengan menggunakan kriteria stabilitas *Nyquist*. Untuk nominal sistem loop terbuka $G_m(j\omega)C(j\omega)$ dapat didefinisikan secara matematis sebagai berikut :

$$\pi = \{G: |G(j\omega) - G_m(j\omega)| \leq \pi) \bar{l}_a(\omega)\} \quad (2.18)$$

Kemudian juga didefinisikan bahwa

$$|l_a(j\omega)| \leq \bar{l}_a(\omega) \quad (2.19)$$

Jika ketidakpastian *multiplicative* didefinisikan sebagai berikut

$$l_m(j\omega) = \frac{l_a(j\omega)}{G_m(j\omega) - C(j\omega)} \quad (2.20)$$

Dan batasan dari ketidakpastian *multiplicative* menjadi persamaan sebagai berikut

$$\bar{l}_m(\omega) = \frac{\bar{l}_a(\omega)}{|G_m(j\omega) - C(j\omega)|} \quad (2.21)$$

Dari persamaan 2.21 batasan ketidakpastian menjadi

$$\bar{l}_a(\omega) = |G_m(j\omega) - C(j\omega)| \bar{l}_m(\omega) \quad (2.22)$$

Kestabilan *robust* dapat dikatakan jika semua plant $G(s)$ dalam keluarga π memiliki jumlah yang sama dengan pole RHP dan sebuah controller tertentu $C(s)$ menstabilkan plant nominal $G_m(j\omega)$, maka sistemnya stabil dengan kontroler $C(s)$ jika dan hanya jika fungsi sensitifitas komplemen $T(s)$ untuk plant nominal $G_m(s)$ memenuhi batasan berikut :

$$\|T(j\omega)\bar{l}_m(\omega)\|_\infty = \sup_{\omega} |T(j\omega)\bar{l}_m(\omega)| < 1 \quad (2.23)$$

Hal ini menunjukkan bahwa kestabilan *robust* memaksakan sebuah keterikatan pada H_∞ dari fungsi sensitivitas komplemen $T(j\omega)$ dengan pembobot $\bar{l}_m(\omega)$.

2.4.3 Performansi H_2 dan H_∞

Untuk performansi H_2 dan H_∞ digunakan untuk mendesain kontrol yang tepat (Σ_c) sehingga ketika dimasukan ke plant (Σ) dengan gangguan, dapat diperoleh sistem lup tertutup yang stabil secara internal dan fungsi alih lup tertutup dari gangguan w terhadap output yang dikontrol z , T_{zw} dapat sekecil mungkin sehingga pengaruh gangguan terhadap keluaran yang dikontrol akan minimum (Erwin Susanto, 2015).

Dari Gambar 2.7 dapat dibuat *generaliz plant* sebagai berikut :

$$\Sigma : \dot{x} = Ax + Bu + Ew \quad (2.24)$$

$$y = C_1x + 0u + D_1w \quad (2.25)$$

$$z = C_2x + D_2u + 0w \quad (2.26)$$

Sistem dinamik kontroler dapat dituliskan dalam bentuk persamaan berikut :

$$\Sigma_c : \dot{v} = A_c v + B_c y \quad (2.27)$$

$$u = C_c v + D_c y \quad (2.28)$$

Dimana x adalah variabel *state*, v adalah kontroler *state*.

Kemudian persamaan 2.25 disubstitusi ke 2.27, sehingga

$$\dot{v} = A_c v + B_c (C_1x + D_1w) = A_c v + B_c C_1x + B_c D_1w \quad (2.29)$$

Selanjutnya persamaan 2.28 disubstitusikan ke persamaan 2.24 dan 2.26 sehingga \dot{x}, y, z menjadi:

$$\dot{x} = Ax + B(C_c v + D_c y) + Ew \quad (2.30)$$

$$y = C_1x + 0u + D_1w \quad (2.31)$$

$$z = C_2x + D_2(C_c v + D_c y) \quad (2.32)$$

Sehingga jika persamaan 2.31 dimasukkan ke persamaan 2.30 dan 2.32 persamaan menjadi sebagai berikut:

$$\dot{x} = Ax + BC_c v + BD_c(C_1x + D_1w) + Ew \quad (2.33)$$

$$z = C_2x + D_2C_c v + D_2D_c(C_1x + D_1w) \quad (2.34)$$

Persamaan 2.33 dan 2.34 dapat dikelompokkan menjadi

$$\dot{x} = (A + BD_cC_1)x + BC_c v + (E + BD_cD_1)w \quad (2.35)$$

$$z = (C_2 + D_2D_cC_1)x + D_2C_c v + D_2D_cD_1w \quad (2.36)$$

Apabila disusun dalam bentuk matriks persamaan 2.35 dan 2.29 menjadi

$$\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{bmatrix} A + BD_c C_1 & BC_c \\ B_c C_1 & A_c \end{bmatrix} \begin{pmatrix} x \\ v \end{pmatrix} + \begin{bmatrix} E + BD_c D_1 \\ B_c D_1 \end{bmatrix} w \quad (2.37)$$

$$z = [C_1 + D_2 D_c C_1 \quad D_2 D_2 C_c] \begin{pmatrix} x \\ v \end{pmatrix} = D_2 D_c D_1 w \quad (2.38)$$

Dimana persamaan 2.37 dan 2.38 dapat ditulis menjadi

$$\begin{pmatrix} \dot{\tilde{x}} \\ \dot{v} \end{pmatrix} = A_{cl} \tilde{x} + B_{cl} w \quad (2.39)$$

$$z = C_{cl} \tilde{x} + D_{cl} w \quad (2.40)$$

Sehingga fungsi alih closed lopp w - z dituliskan sebagai berikut :

$$T_{zw}(s) = C_{cl}(sI - A_{cl}) - 1B_{cl} + D_{cl} \quad (2.41)$$

Sistem closed loop akan stabil secara internal jika dan hanya jika nilai eigen dari $A_{cl} = \begin{bmatrix} A + BD_c C_1 & BC_c \\ B_c C_1 & A_c \end{bmatrix}$ terletak disebelah kiri bidang kompleks.

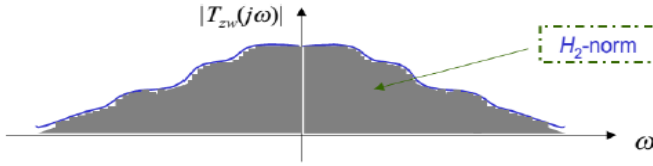
Fungsi alih memiliki frekuensi berkisar dari 0 sampai ∞ . Sehingga cukup sulit mengatakan pengaruh tersebut besar atau kecil. Metode praktis yang dapat digunakan yaitu dengan mengukur nilai norm nya yaitu H_2 - norm dan H_∞ - norm.

- *Norm H_2*

Norm H_2 merupakan energy keseluruhan berkaitan dengan tegangan implus dari $T_{zw}(s)$ sehingga meminimalkan norm H_2 suatu fungsi alih $T_{zw}(s)$ berarti meminimalkan energy dari gangguan w menuju output terkontrol z . Untuk fungsi alih $T_{zw}(s)$ yang stabil dan sesuai, maka norm H_2 nya didefinisikan sebagai berikut:

$$\|T_{zw}\|_2 = \left(\left(\frac{1}{2\pi} \right) \text{trace} \left[\int_{-\infty}^{\infty} T_{zw}(j\omega) T_{zw}^*(j\omega) d\omega \right] \right)^{\frac{1}{2}} \quad (2.42)$$

Secara grafis dapat digambarkan sebagai berikut



Gambar 2. 12 Ilustri grafik luasan H_2 -norm

Infimum norm H_2 - suatu matriks alih closed loop $T_{zw}(s)$ yang secara keseluruhan menstabilkan kontrol yang tepat dapat dituliskan :

$$\gamma_2 : \inf \{ \|T_{zw}\|_2 \mid \Sigma \text{ internally stabilizes } \Sigma \} \quad (2.43)$$

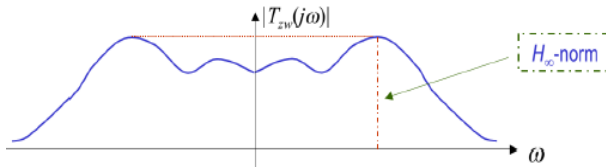
Kontrol Σ_c dikatakan sebagai kontrol optimal H_2 jika secara internal menstabilkan Σ dan $\|T_{zw}\| = \gamma_2$ (John Doyle, 1990)

- *Norm H_∞*

Norm ini merupakan penguatan untuk kondisi terburuk dalam $T_{zw}(s)$ sehingga meminimalkan norm H_∞ suatu fungsi alih $T_{zw}(s)$ berarti meminimalkan situasi penguatan terburuk dari gangguan w menuju output terkontrol z . Untuk fungsi alih $T_{zw}(s)$ yang stabil dan sesuai, maka norm H_∞ nya didefinisikan sebagai berikut:

$$\begin{aligned} \|T_{zw}\|_\infty &= \sup \sigma_{\max}[T_{zw}(j\omega)] \\ \|T_{zw}\|_\infty &= \sigma_{\max} \end{aligned} \quad (2.44)$$

Secara grafis dapat digambarkan sebagai berikut



Gambar 2. 13 Ilustri grafik luasan H_∞ -norm

Infimum norm H_∞ suatu matriks alih closed loop $T_{zw}(s)$ yang secara keseluruhan menstabilkan kontrol yang tepat ,dituliskan

$$\gamma_2: \inf\{\|T_{zw}(s)\|_\infty \mid \Sigma_c \text{ internally stabilizes } \Sigma\} \quad (2.45)$$

Kontrol Σ_c dikatakan sebagai kontrol optimal H- jika secara internal menstabilkan Σ dan $\|T_{zw}\|_\infty < \gamma (> \gamma_-)$

2.5 Sistem *Passive Fault Tolerant Control (PFTC)*

Dalam suatu sistem kontrol, dibutuhkan suatu peningkatan performa dan keamanan yang cukup tinggi. Dengan menggunakan *design feedback* kontrol dapat mengakibatkan kinerja yang tidak memuaskan sehingga terkadang terjadi kerusakan pada sensor, aktuator ataupun komponen-komponen yang menyusun sistem tersebut. Untuk mengatasi kerusakan tersebut, dilakukan desain kontrol yang mampu mentolerir kerusakan yang terjadi agar stabilitas sistem tetap terjaga sesuai yang diinginkan. Sistem *Fault Tolerant Control* (FTC) adalah suatu sistem kontrol yang mampu mengatasi terjadinya kegagalan secara otomatis dengan menjaga stabilitas dan kinerja yang dapat diterima dalam suatu kegagalan (Zhang & Jiang, 2008).

FTCS dapat diklasifikasikan pada 2 tipe, yaitu *Passive Fault Tolerant Control Sistem* (PFTCS) dan *Active Fault Tolerant Control Sistem* (AFTCS) (Zhang & Jiang, 2008). PFTCS dirancang untuk mempertahankan sistem dari kegagalan yang terjadi. Sedangkan AFTCS bereaksi terhadap kegagalan komponen dengan merekonfigurasi aksi kontrol sehingga kestabilan dan kinerja pada sistem dapat dijaga. Perbedaan antara sistem PFTC dan AFTC adalah terletak pada perancangan *Reconfigurable Control* (RC) dan FDD dimana perancangan tersebut diperlukan dalam sistem AFTC sedangkan sistem PFTC tidak memerlukan perancangan tersebut.

Pada *Passive Fault Tolerant Control (PFTC)* parameter kontroler dibuat tetap dan didesain menggunakan algoritma kontrol *robust* untuk memastikan sistem kontrol mampu mengatasi

kesalahan yang terjadi baik pada komponen sensor maupun aktuator (Zhao, 1999). Kelebihan sistem PFTC yaitu karena besarnya kesalahan sudah dapat ditentukan sebelumnya sehingga sistem dapat didesain menjadi sistem pengendali yang *robust* terhadap kesalahan yang terjadi ada suatu komponen dan tidak terjadi penundaan waktu karena tidak memerlukan prosedur RC dan FDD seperti pada sistem AFTC. Namun konsekuensinya, jika besaran kesalahan diluar nilai yang ditentukan maka kinerja sistem ini akan memburuk. Oleh karena itu dalam perancangan sistem PFTC perlu adanya *observer* untuk mengestimasi besaran kesalahan sehingga sistem dapat kestabilan dan performansi dapat terjaga.

2.5.1 *Observer*

Observer adalah suatu algoritma yang digunakan untuk mengestimasi keadaan (*state*) dari variabel yang tidak terukur berdasarkan model matematis sistem sehingga *observer* juga dapat digunakan untuk mengestimasi kesalahan yang terjadi pada suatu sistem jika kesalahan tersebut masuk dalam pemodelan (Zhang & Jiang, 2008) (Indriawati, 2015). Suatu sistem umumnya dimodelkan dalam bentuk *state space*, yang dapat didefinisikan secara umum seperti persamaan sistem sebagai berikut:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.44)$$

$$y(t) = Cx(t) + Du(t) \quad (2.45)$$

Persamaan ruang keadaan yang mengandung kesalahan sensor dituliskan sebagai berikut :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.46)$$

$$y(t) = Cx(t) + Du(t) + Ff_s(t) \quad (2.47)$$

Dengan $x(t) \in R^n$, $u(t) \in R^m$, $y(t) \in R^p$, $f_s(t) \in R^q$ berturut – turut adalah keadaan (*state*), masukan, keluaran, dan kesalahan

sensor. Sedangkan A, B, C, F_s adalah matriks keadaan, masukan, keluaran, dan kesalahan sensor.

Kemudian *state* baru ditambahkan untuk memindahkan kesalahan sensor dari persamaan keluaran ke persamaan keadaan

$$\dot{x}_z = A_z (y - x_z) \quad (2.48)$$

Persamaan 2.47 disubstitusi ke dalam persamaan 2.48 sehingga diperoleh persamaan:

$$\dot{x}_z = A_z (Cx + Du + F_s f_s - x_z) \quad (2.49)$$

Untuk A_z merupakan matriks identitas (I) untuk $A_z \in I^q$, sedangkan matriks dari kesalahan sensor didefinisikan sebagai $f = f_s$

Dengan mendefinisikan variable keadaan diperbanyak atau *augmented state* maka diperoleh persamaan ruang keadaan baru :

$$\dot{x}_a = A_a x_a + B_a u + E_a f \quad (2.50)$$

$$y_a = C_a x_a \quad (2.51)$$

Sehingga persamaan 2.50 dan 2.51 dapat diubah ke dalam bentuk persamaan yang lebih kompak menjadi

$$\begin{bmatrix} \dot{x} \\ \dot{x}_z \end{bmatrix} = \begin{bmatrix} A & 0 \\ A_z C & -A_z \end{bmatrix} \begin{bmatrix} x \\ x_z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ A_z F_s \end{bmatrix} [f_s] \quad (2.52)$$

$$y = \dot{x}_z = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} x \\ x_z \end{bmatrix} \quad (2.53)$$

dengan

$$x_a = \begin{bmatrix} x \\ x_z \end{bmatrix}, \quad A_a = \begin{bmatrix} A & 0 \\ A_z C & -A_z \end{bmatrix}, \quad B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad C_a = \begin{bmatrix} 0 & I \end{bmatrix}$$

$$E_a = \begin{bmatrix} 0 \\ A_z F_s \end{bmatrix}$$

Dari persamaan 2.50 dan 2.51 dapat diekspansi menjadi persamaan dinamika *observer* :

$$\dot{\hat{x}}_a = A_a \hat{x}_a + B_a u + E_a \hat{f}_s + L_x (Y - \hat{Y}) \quad (2.54)$$

$$\dot{\hat{f}}_s = L_f (Y - \hat{Y}) \quad (2.55)$$

$$\hat{y}_a = C_a \hat{x}_a \quad (2.56)$$

Dengan $\hat{x}_a, \hat{f}_a, \hat{y}$ berturut turut adalah hasil estimasi *augmented state*, kesalahan dan output. L_x adalah gain proporsional dari *observer* dan L_f adalah gain integral dari *observer*. Gain *observer* ini digunakan untuk menjamin kestabilan dinamik dari *error* estimasi.

2.6 Optimisasi dengan LMI (*Linier Matrikx Ineequalities*)

LMI (*Linier Matrikx Ineequalities*) yang merupakan perhitungan numeric menggunakan metode optimasi konveks untuk menyelesaikan petidaksamaan matriks. LMI memiliki bentuk umum sebagai berikut:

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.57)$$

Dengan $x = (x_1, \dots, x_N)$ adalah vector scalar yang tidak diketahui (variabel optimisasi) dan F_i adalah matriks simetris yang diketahui.

Problem optimisasi didefinisikan dengan LMI yakni meminumkan sebuah *cost function* yaitu fungsi x yang hasilnya pasti skalar, dan LMI sendiri sebagai *constraint* atau fungsi kendala Dengan meminimumkan nilai x sedemikian hingga mencapai stabil

atau hingga $F(x) > 0$ maka memenuhi untuk kasus kestabilan *Lyapunov*.

Berdasarkan analisa kestabilan *Lyapunov*, kestabilan didasarkan pada energy yang disimpan oleh sistem. Sistem dapat dikatakan stabil jika energy yang disimpan berkurang terhadap pertambahan waktu, sehingga energy akan mencapai pada nilai minimal pada titik setimbang sistem. Fungsi *Lyapunov* adalah fungsi energy buatan yang tergantung pada vector state sistem ($x = [x_1 x_2 \dots x_n]^T$) dapat dinyatakan dengan $V(x)$. Dalam metode kedua analisa kestabilan *Lyapunov*, $V(x)$ dan turunan pertamanya $\dot{V}(x)$ dapat memberikan informasi kestabilan sistem (Modern Control Engineering, 1981).

Untuk kasus pertidaksamaan *Lyapunov* dapat dituliskan dengan persamaan berikut :

$$P > 0 \quad (2.58)$$

$$A^T P + PA < 0 \quad (2.59)$$

Dimana P adalah matriks simetris, sehingga untuk mencapai stabil maka nilai LMI menjadi :

$$\bar{F} = \begin{bmatrix} -P & 0 \\ 0 & A^T P + PA \end{bmatrix} \quad (2.60)$$

Kegunaan utama LMI adalah kemampuan untuk menggabungkan berbagai fungsi kendali (*constraint*) atau tujuan desain secara numeric.

Untuk kriteria kestabilan dapat dicari nilai LMI dengan contoh kasus sebagai berikut :

$$\text{Kasus} \quad \dot{x} = Ax + Bu \quad (2.61)$$

$$u = Kx \quad (2.62)$$

Dari persamaan 2.62 disubstitusi ke persamaan 2.61 sehingga menjadi

$$\begin{aligned}
\dot{x} &= Ax + BKx \\
\dot{x} &= (A + BK)x \\
\dot{x} &= A_c x
\end{aligned} \tag{2.63}$$

Dengan memenuhi syarat kestabilan *Lyapunov* maka

$$\begin{aligned}
\dot{v} &< 0 \rightarrow P > 0 \\
A_c^T P + PA
\end{aligned} \tag{2.64}$$

Dari persamaan 2.63 dapat disubstitusikan ke persamaan 2.63 menjadi

$$\begin{aligned}
&(A + BK)^T P + P(A + BK) \\
&= A^T P + (BK)^T P + PA + PBK \\
&= A^T P + K^T B^T P + PA + PBK \\
&= A^T P + PA + K^T B^T P + PBK
\end{aligned} \tag{2.65}$$

Maka didapatkan nilai LMI sebagai berikut :

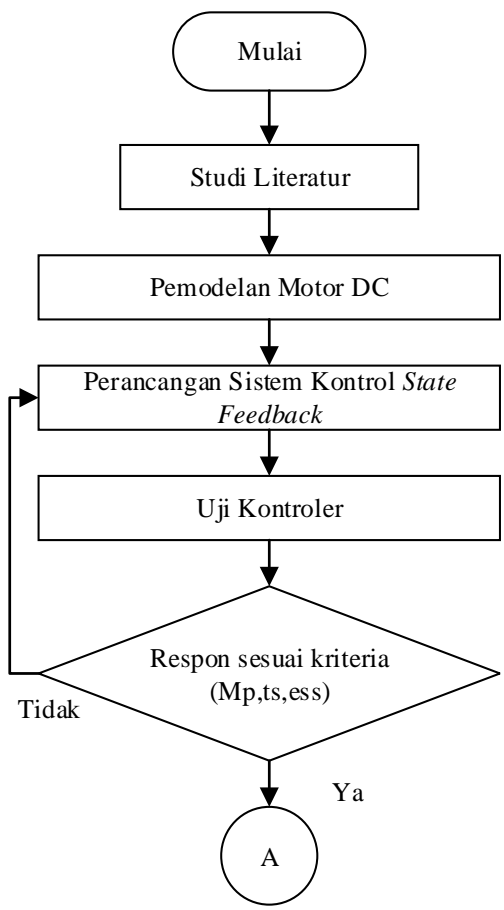
$$\begin{aligned}
LMI &= \begin{bmatrix} -P & 0 \\ 0 & A_c^T P + PA_c \end{bmatrix} \\
LMI &= \begin{bmatrix} -P & 0 \\ 0 & A^T P + PA + K^T B^T P + PBK \end{bmatrix}
\end{aligned} \tag{2.66}$$

Nilai LMI pada saat digunakan untuk mencari nilai *observer* seringkali tidak terbaca di Matlab sehingga diperlukan perhitungan menggunakan *Schur Complement*.

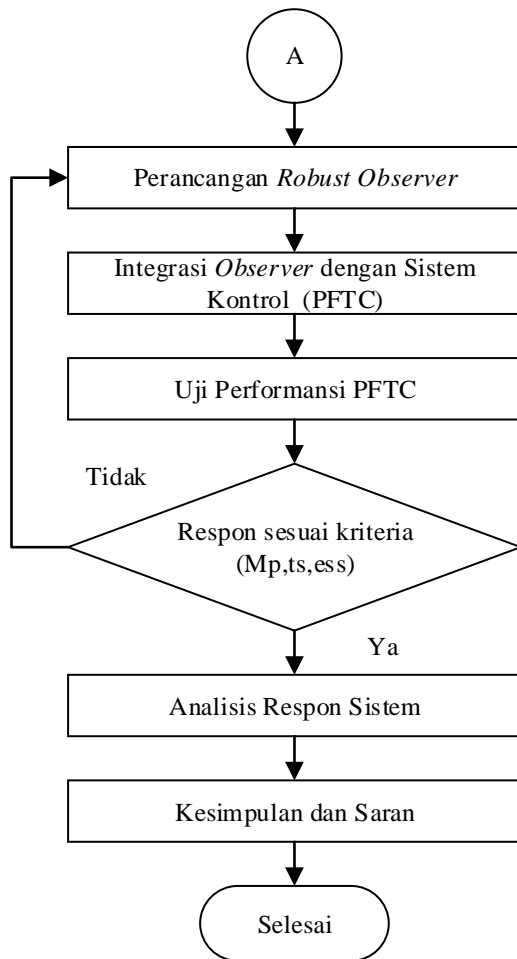
BAB III

METODOLOGI PENELITIAN

Metodologi yang digunakan untuk mencapai tujuan dari tugas akhir ini dijelaskan melalui diagram alir yang ditunjukkan pada Gambar 3.1 yang mendeskripsikan tahap-tahap penelitian yang dimulai dari studi literature tentang sistem servo motor DC, kontrol *state feedback*, kontrol *robust*, sistem *Passive Fault Toleran Control* dan *Observer*. Kemudian dilanjutkan dengan pemodelan motor DC mulai dari sensor *tachometer* dan aktuator berupa *servo amplifier* untuk mendapatkan persamaan ruang keadaan dari motor DC. Selanjutnya dilakukan perancangan kontroller *state feedback* untuk mendapatkan nilai *gain* controller. Untuk memastikan respon sesuai kriteria maka dilakukan uji kontroler. Setelah itu dilakukan perancangan *robust observer* untuk mendapatkan *gain observer* yang dapat mengestimasi besar kesalahan. Setelah *gain observer* didapatkan maka dilakukan integrasi *observer* dengan plant kontrol (PFTC). Uji performansi PFTC dilakukan dengan memberikan kesalahan bias dan sensitifitas untuk sensor sehingga dapat dianalisis respon apakah sesuai dengan kriteria.



Gambar 3. 1 Diagram Alir Peneluitian Tugas Akhir



Gambar 3.1 (Lanjutan)

3.1 Pemodelan Sistem Motor DC

- *Plant* Motor DC

Dari persamaan 2.14 diperoleh model matematik dari motor DC dengan beberapa parameter Berikut nilai dari masing-masing parameter sistem motor DC (Katherin I, 2015):

$$J = 0,01, b = 0,1, K_e = 0,01, R = 1, L = 0,5$$

Dari parameter motor DC tersebut, dapat dicari fungsi transfer motor DC dengan menggunakan persamaan 2.14 sebagai berikut :

$$\frac{\omega(s)}{V(s)} = \frac{K}{RJs + Rb + K.Kb} \quad (3.1)$$

$$\frac{\omega(s)}{V(s)} = \frac{0.01}{1s + 1(0.1) + (0.01)(0.01)} \quad (3.2)$$

$$\frac{\omega(s)}{V(s)} = \frac{0.01}{s + 0.1 + 0.0001} \quad (3.3)$$

Dari fungsi transfer tersebut dapat diketahui bahwa nilai *time konstan* (τ) motor DC sebesar 0.01.

Kemudian dari fungsi transfer tersebut disusun dalam bentuk persamaan ruang keadaan untuk sistem servo motor DC sesuai dengan persamaan 2.12 dan 2.13, menjadi sebagai berikut

$$\begin{bmatrix} \dot{\omega} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} -10 & 1.0 \\ -0.02 & -2.0 \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} V \quad (3.4)$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} \quad (3.5)$$

- *Sensor Tachometer*

Sensor yang digunakan untuk mengukur kecepatan dari motor DC yaitu *tachometer* tipe BD2510B (Lihat di lampiran) . Dari data spesifikasi *tachometer* tipe BD2510B didapatkan nilai *time konstan* sebesar $5,45 \times 10^{-5} s$. Nilai *time konstan* sensor jauh lebih cepat daripada *time konstan plant* maka *time konstan* sensor dapat diabaikan. Sehingga kecepatan yang diukur memiliki memiliki

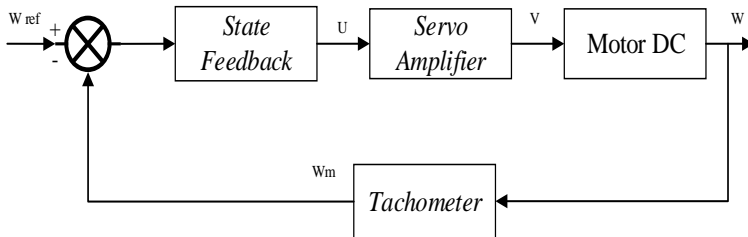
dinamika yang sama dengan kecepatan plant atau dapat dituliskan bahwa nilai $\omega = \omega_m$.

- *Servo amplifier*

Aktuator yang digunakan untuk mengendalikan motor DC adalah *servo amplifier*. *Servo amplifier* tersebut memiliki *timekonstan* sebesar 0,5 ms yang lebih cepat daripada *time konstan* plant sehingga *time constan servo amplifier* dapat diabaikan (H.Kohlmann, 2015). Tegangan yang diukur memiliki memiliki dinamika yang sama dengan tegangan plant sehingga dapat dituliskan bahwa nilai $u = v$.

3.2 Perancangan Sistem Kontrol *State Feedback*

Sistem motor DC terdiri atas beberapa komponen yaitu *plant* motor DC, sensor *tachometer* dan actuator *servo amplifier*, sehingga pemodelannya meliputi itu semua. Secara sistem *close loop* dapat ditunjukkan dengan diagram nlok sebagai berikut :



Gambar 3. 2 Diagram Blok Sistem Pengendalian Kecepatan Motor

Persamaan ruang keadaan sistem atau *state space* didefinisikan dengan persamaan berikut :

$$\dot{x} = Ax + Bu \quad (3.6)$$

$$y = Cx + Du \quad (3.7)$$

Dalam sistem motor DC ini memiliki *set point* berupa kecepatan (ω) dan arus (i) sehingga dapat dituliskan sebagai berikut :

$$x = \begin{bmatrix} \omega \\ i \end{bmatrix} \quad (3.8)$$

Sedangkan output yang diinginkan mengikuti *set point* dari sistem motor DC adalah kecepatan (ω) sehingga

$$y_r = C_r \cdot x = [1 \quad 0] \begin{bmatrix} \omega \\ i \end{bmatrix} = \omega \quad (3.9)$$

Didefinisikan *error tracking* adalah perbedaan antara *set point* dengan pengukur yang dinyatakan dalam persamaan berikut :

$$\dot{e} = \omega_{ref} - \omega \quad (3.10)$$

$$\dot{e} = \omega_{ref} - C_r \cdot x \quad (3.11)$$

$$\dot{e} = -\omega_{ref} + C_r \cdot x \quad (3.12)$$

Dengan menggabungkan persamaan 3.6 dan 3.12 maka dibentuk persamaan *state space* baru sebagai berikut :

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C_r & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \omega_{ref} \quad (3.13)$$

Sehingga dapat dituliskan kembali menjadi

$$\dot{x}_c = A_c x_c + B_c u \quad (3.14)$$

$$\dot{x}_c = (A_c + B_c K_c) x_c \quad (3.15)$$

Dimana $K_c = [K \quad K_i]$

Diinginkan respon maksimum *overshoot*, *time settling* dan *error steady state* yang kecil agar sistem yang dihasilkan stabil, maka ditentukan *pole* yang diinginkan sebagai berikut:

$$p1 = -10 + i$$

$$p2 = -10 - i$$

$$p_3 = -3$$

Nilai *gain* kontroler ditentukan dengan menggunakan metode *pole placement* sehingga diperoleh :

$$K_c = [K \quad K_i] \quad (3.16)$$

$$K_c = [15.4900 \quad 5.5000 \quad 151.5000] \quad (3.17)$$

3.3 Perancangan *Observer*

Perancangan *observer* digunakan untuk mengestimasi kesalahan yang ada dengan cara mengembangkan persamaan ruang keadaan sistem. Persamaan ruang keadaan yang mengandung kesalahan sensor dituliskan sebagai berikut :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.18)$$

$$y(t) = Cx(t) + Du(t) + F_s f_s(t) \quad (3.19)$$

Dengan $x(t) \in R^n, u(t) \in R^m, y(t) \in R^p, f_s(t) \in R^q$ berturut –turut adalah keadaan (*state*), masukan, keluaran, dan kesalahan sensor. Sedangkan A, B, C, F_s adalah matriks keadaan, masukan, keluaran, dan kesalahan sensor.

Tipe kesalahan sensor dimodelkan sebagai

$$s_f(t) = as(t) + \beta \quad (3.20)$$

$$f(t) = s_f(t) - s(t) \quad (3.21)$$

Dengan $s_f(t)$ adalah sinyal actual, α dan β adalah konstanta.

Kemudian *state* baru ditambahkan untuk memindahkan kesalahan sensor dari persamaan keluaran ke persamaan keadaan

$$\dot{x}_z = A_z(y - x_z) \quad (3.22)$$

Persamaan 3.19 disubstitusi ke dalam persamaan 3.22 sehingga diperoleh persamaan:

$$\dot{x}_z = A_z(Cx + Du + F_s f_s - x_z) \quad (3.23)$$

Untuk A_z merupakan matriks identitas (I) untuk $A_z \in I^q$, sedangkan matriks dari kesalahan sensor didefinisikan sebagai $f = f_s$

Dengan mendefinisikan variable keadaan diperbanyak atau *augmented state* maka diperoleh persamaan ruang keadaan baru :

$$\dot{x}_a = A_a x_a + B_a u + E_a f \quad (3.24)$$

$$y_a = C_a x_a \quad (3.25)$$

Sehingga persamaan 3.18 dan 3.23 dapat diubah ke dalam bentuk persamaan yang lebih kompak menjadi

$$\begin{bmatrix} \dot{x} \\ \dot{x}_z \end{bmatrix} = \begin{bmatrix} A & 0 \\ A_z C & -A_z \end{bmatrix} \begin{bmatrix} x \\ x_z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ A_z F_s \end{bmatrix} [f_s] \quad (3.26)$$

$$y = \dot{x}_z = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} x \\ x_z \end{bmatrix} \quad (3.27)$$

dengan

$$x_a = \begin{bmatrix} x \\ x_z \end{bmatrix}, A_a = \begin{bmatrix} A & 0 \\ A_z C & -A_z \end{bmatrix}, B_a = \begin{bmatrix} B \\ 0 \end{bmatrix}, C_a = \begin{bmatrix} 0 & I \end{bmatrix}, E_a = \begin{bmatrix} 0 \\ A_z F_s \end{bmatrix}$$

Dari persamaan 3.24 dan 3.25 dapat diekspansi menjadi persamaan dinamika *observer* berikut :

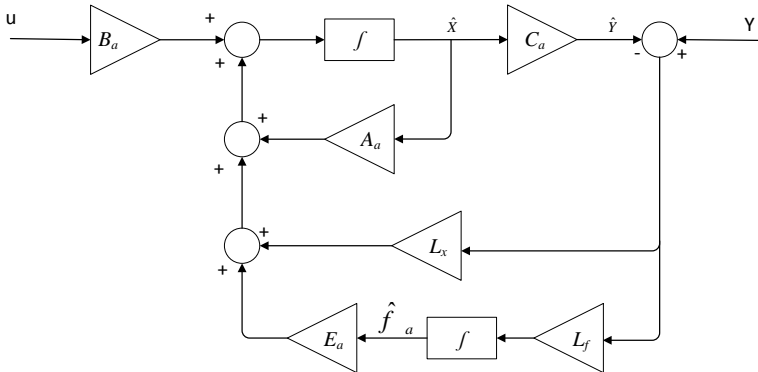
$$\dot{\hat{x}}_a = A_a \hat{x}_a + B_a u + E_a \hat{f}_s + L_x (Y - \hat{Y}) \quad (3.28)$$

$$\dot{\hat{f}}_s = L_f (Y - \hat{Y}) \quad (3.29)$$

$$\hat{y}_a = C_a \hat{x}_a \quad (3.30)$$

Dengan $\hat{x}_a, \hat{f}_a, \hat{y}$ berturut turut adalah hasil estimasi *augmented state*, kesalahan dan output. L_x adalah gain proporsional dari *observer* dan L_f adalah gain integral dari

observer. Gain *observer* ini digunakan untuk menjamin kestabilan dinamik dari *error* estimasi.



Gambar 3.3 Struktur *State Observer*

3.3.1 Penentuan *Generalized Plant*

Tahap selanjutnya adalah mencari nilai *gain observer* supaya *robust* terhadap *disturbance* (gangguan) yang dapat disebut dengan *robust observer*. Metode yang digunakan untuk mencari *gain* tersebut adalah dengan menggunakan LMI (*Linier Matrx Ineeequalities*). Prosedur awal yang dilakukan adalah menentukan *generalized plant* dari persamaan 2.24 untuk mengakomodasi *disturbance* sehingga diperoleh nilai t_a , t_b , t_c , dan t_d .

Penjabaran dari model *generalized plant* dijelaskan sebagai berikut :

Untuk kontroler diberikan input berupa y_{ref} dan \hat{x} dengan output yaitu u , sehingga dapat ditulis dalam bentuk persamaan sebagai berikut :

$$u_c = \begin{bmatrix} y_{ref} \\ \hat{x} \end{bmatrix} \quad (3.31)$$

Jika didefinisikan persamaan ruang keadaan *state space* kontroler sebagai berikut :

$$\dot{x}_c = A_c x_c + B_c u_c \quad (3.32)$$

$$y_c = u \quad (3.33)$$

$$y_c = C_c x_c + D_c u_c \quad (3.34)$$

Didapatkan nilai *error set point* dengan persamaan berikut :

$$\dot{e}_c = y_{ref} - C_r \hat{x} \quad (3.35)$$

Kemudian \dot{e}_c didefinisikan sama dengan *state* kontroler \dot{x}_c sehingga dapat ditulis dalam bentuk persamaan sebagai berikut :

$$\dot{x}_c = -C_r \hat{x} + y_{ref} \quad (3.36)$$

Sedangkan untuk nilai *output* dengan adanya nilai *gain* K_p dan K_i maka dapat diperoleh persamaan sebagai berikut

$$-e_c K_i - K_p \hat{x} = u \quad (3.37)$$

$$-e_c K_i - K_p \hat{x} = y_c \quad (3.38)$$

$$y_c = -K_i e_c - K_p \hat{x} \quad (3.39)$$

Dari persamaan input dan output kontroler didapatkan nilai matriks sebagai berikut :

$$A_c = 0, B_c = [1 \quad -C_r], C_c = -K_i, D_c = [0 \quad -K_p]$$

Didefinisikan persamaan ruang keadaan dari *plant* sebagai berikut :

$$\dot{x} = Ax + Bu \quad (3.40)$$

Dengan mensubstitusikan persamaan 3.37 ke persamaan 3.40 maka didapatkan persamaan sebagai berikut :

$$\dot{x} = Ax + B(-K_i e_c - K_p \hat{x}) \quad (3.41)$$

$$\dot{x} = Ax - BK_i e_c - BK_p \hat{x} \quad (3.42)$$

Pada pengukuran atau *measurement* , representasi kesalahan dinyatakan dengan nilai fgs . Jika $fgs=1$ atau tanpa kesalahan maka $\omega_m = \omega$. Karena besaran yang diukur adalah kecepatan atau ω_m maka dapat ditulis dalam bentuk persamaan berikut :

$$\omega_m = F_s \omega \quad (3.43)$$

Apabila di ubah dalam bentuk persamaan ruang keadaan maka diperoleh sebagai berikut :

$$\begin{bmatrix} \omega_m \\ i \end{bmatrix} = \begin{bmatrix} F_s & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \omega \\ i \end{bmatrix} \quad (3.44)$$

Sehingga output yang dihasilkan dari pengukuran adalah

$$y_m = F_s Cx \quad (3.45)$$

Output pengukuran atau *measurement* y_m akan menjadi inputan untuk *observer*. *Observer* juga mendapat inputan dari kontroler dengan simbol u . Dengan mengambil dari struktur *observer* didapatkan tiga *state* yaitu $\hat{x}_z, \hat{f}_s, \hat{x}$ dengan persamaan sebagai berikut :

$$\dot{\hat{x}}_z = A_z(Cx + Du + F_s f_s - x_z) \quad (3.46)$$

$$\dot{\hat{f}}_s = L_f e_L \quad (3.47)$$

$$\dot{\hat{x}} = A\hat{x} + Bu + L_x e_L \quad (3.48)$$

Didefinisikan *error* residual dari *observer* dengan persamaan sebagai berikut :

$$e_{xz} = x_z - \hat{x}_z \quad (3.49)$$

Dimana

$$\hat{x}_z = C_a \hat{x}_a \quad (3.50)$$

Output performansi merupakan output yang akan dibuat acuan yang mewakili unjuk kerja dari sitem. Output performansi terdiri dari output *error tracking* dan output *error residual*. Sehingga output performansi atau e_z , dapat ditulis dalam bentuk persamaan sebagai berikut :

$$e_z = y_r - C_r x \quad (3.51)$$

Sehingga

$$\dot{e}_z = \dot{y}_r - C_r \dot{x} \quad (3.52)$$

Apabila \dot{y}_r dianggap konstan = 0 maka persamaan 3.52 menjadi sebagai berikut :

$$\dot{e}_z = -C_r (Ax + Bu) \quad (3.53)$$

Didefinisikan output performansi adalah z yang memiliki dengan hubungan *error tracking* sehingga dalam bentuk fungsi transfer dapat dituliskan dalam persamaan berikut :

$$W(s) = \frac{z(s)}{e_c(s)} = \frac{A_1 s + A_2}{s + A_3} \quad (3.54)$$

Persamaan 3.57 dijabarkan sehingga menjadi persamaan berikut ini :

$$\dot{z} + A_3 z = A_1 \dot{e}_c + A_2 e_c \quad (3.55)$$

Kemudian persamaan 3.35 disubstitusikan ke persamaan 3.55 sehingga persamaan menjadi

$$\dot{z} = -A_3 z + A_1 (y_{ref} - C_r \hat{x}) + A_2 e_c \quad (3.56)$$

$$\dot{z} = -A_3 z + A_1 y_{ref} - A_1 C_r \hat{x} + A_2 e_c \quad (3.57)$$

Persamaan ruang keadaan untuk *generalized plant* dengan menggunakan enam *state* yaitu x_z , \hat{f}_s , \hat{x} , x , e_c dan z sehingga dapat dituliskan dengan persamaan sebagai berikut :

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{f}} \\ \dot{\hat{x}} \\ \dot{x} \\ \dot{e}_c \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -I & AF & 0 & AC & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -BK_r + A & 0 & -BK_r & 0 \\ 0 & 0 & -BK_r & A & -BK_r & 0 \\ 0 & 0 & -C_r & 0 & 0 & 0 \\ 0 & 0 & -AC_r & 0 & A & -A_r \end{bmatrix} \begin{bmatrix} x \\ \hat{f} \\ \hat{x} \\ x \\ e_c \\ z \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & L_f \\ 0 & L_x \\ 0 & 0 \\ I & 0 \\ A_1 & 0 \end{bmatrix} \begin{bmatrix} y_{ref} \\ e_L \end{bmatrix} \quad (3.58)$$

Dari persamaan ruang keadaan *generaliz plant* tersebut didapatkan nilai matriks ta,tb,tc,dan td (Lampiran).

3.3.2 Penentuan Gain Observer

Setelah penentuan *generalized plant* maka dilakukan penentuan *gain observer* dengan metode optimasi .Problem optimasi dinyatakan dengan meminimumkan γ^2 atau penguatan

supaya subject to γ yang diselesaikan dengan *generalized plant* sebelumnya menggunakan metode dilatasi.

Penjabaran dari metode dilatasi untuk menyelesaikan problem optimasi LMI sebagai berikut :

Persamaan sistem didefinisikan sebagai berikut

$$\dot{x} = Ax \quad (3.59)$$

Dari persamaan (3.59) dapat direpresentasikan *deskriptor* sebagai berikut:

$$\begin{bmatrix} I & O \\ O & O \end{bmatrix} \dot{\tilde{x}} = \begin{bmatrix} O & A \\ I & -I \end{bmatrix} \tilde{x} \quad (3.60)$$

dimana $\tilde{x} = \begin{bmatrix} x^T & x^T \end{bmatrix}^T$. Dengan representasi sistem pada persamaan (3.59) telah dijelaskan bahwa karakterisasi *LMI* yang melebar (dilatasi) atau diperpanjang untuk stabilitas sistem pada persamaan (3.59) dapat diperoleh dengan menerapkan karakterisasi stabilitas untuk sistem deskriptor (Chen G. , 2004)

Karakterisasi dilatasi LMI yang baru membagi matriks A dan B menjadi beberapa bagian, sehingga dapat dituliskan dalam persamaan ruang keadaan sebagai berikut:

$$\dot{x} = Ax + Bu \quad (3.61)$$

$$y = Cx + Du \quad (3.62)$$

Kemudian membagi matriks A dan B menjadi

$$A = \bar{A}_1 + \bar{A}_2 \bar{A}_3^{-1} \bar{A}_4 \quad (3.63)$$

$$B = \bar{B}_1 + \bar{A}_{12} \bar{A}_{22}^{-1} \bar{B}_2 \quad (3.64)$$

Dengan pembagian matriks A dan B tersebut maka representasi untuk *descriptor* dapat dituliskan sebagai berikut

$$\begin{bmatrix} I & O \\ O & O \end{bmatrix} \dot{\tilde{x}} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \tilde{x} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} u \quad (3.65)$$

Dimana $\tilde{x} = \begin{bmatrix} x^T & \xi^T \end{bmatrix}$ dan ξ adalah sebuah *state* (keadaan).

Didefinisikan persamaan *generalized plant* dapat dituliskan sebagai berikut :

$$\dot{x} = Ax + B_0 w_0 + B_1 w_1 + B_2 u \quad (3.66)$$

$$z_0 = C_0 x + D_{01} w_1 \quad (3.67)$$

$$z_1 = C_1 x + D_{10} w_0 + D_{11} w_1 \quad (3.68)$$

Dimana w_0 adalah *set point* dan w_1 *disturbance*.

Dengan nilai *gain* kontroler yang sudah diketahui sebelumnya, maka representasi *descriptor eksplisit* untuk karakterisasi dilatasi LMI, dapat didefinisikan sebagai berikut :

$$\begin{aligned} \tilde{E}\tilde{x} &= \tilde{A}\tilde{x} + \tilde{B}\tilde{w} \\ \tilde{y} &= \tilde{C}\tilde{x} + \tilde{D}\tilde{w} \end{aligned} \quad (3.69)$$

Dengan \tilde{E} = blok diagonal $\{I, O\}$

Sehingga dapat dituliskan dalam bentuk matriks ruang keadaan sebagai berikut

$$\begin{aligned} \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{C} & \tilde{D} \end{bmatrix} &= \begin{bmatrix} \tilde{A} & \tilde{B}_0 & \tilde{B}_1 \\ \tilde{C}_0 & \tilde{D}_{00} & \tilde{D}_{01} \\ \tilde{C}_1 & \tilde{D}_{10} & \tilde{D}_{11} \end{bmatrix} = \begin{bmatrix} A & B_L & B_0 & B_1 \\ O & -I & O & O \\ C_0 & O & O & D_{01} \\ C_1 & O & D_{10} & D_{11} \end{bmatrix} \\ &+ \left\{ \begin{bmatrix} B_2 \\ O \\ O \\ O \end{bmatrix} \hat{K} + \begin{bmatrix} O \\ B_R \\ O \\ O \end{bmatrix} \right\} (K - \hat{K})x \begin{bmatrix} C_2 & O & O & O \end{bmatrix} \quad (3.70) \end{aligned}$$

Kemudian persamaan 3.70 dapat dijabarkan menjadi persamaan sebagai berikut :

$$\begin{aligned}
 & \begin{bmatrix} A & B_L & B_0 & B_1 \\ O & -I & O & O \\ C_0 & O & O & D_{01} \\ C_1 & O & D_{10} & D_{11} \end{bmatrix} + \begin{bmatrix} B_2 \hat{K} C_2 & O & O & O \\ B_R (K - \hat{K}) C_2 & O & O & O \\ O & O & O & O \\ O & O & O & O \end{bmatrix} \\
 &= \begin{bmatrix} A + B_2 \hat{K} C_2 & B_L & B_0 & B_1 \\ B_R (K - \hat{K}) C_2 & -I & O & O \\ C_0 & O & O & D_{01} \\ C_1 & O & D_{10} & D_{11} \end{bmatrix} \quad (3.71)
 \end{aligned}$$

Dimana $B_2 = B_L B_R$, $\xi = B_R U$

$$\tilde{x} = \begin{bmatrix} x \\ \xi \end{bmatrix}, \quad \tilde{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}, \quad \tilde{y} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \quad (3.72)$$

Problem optimasi dinyatakan sebagai berikut :

Minimize γ_2

Subject to $\gamma_2^2 \geq \text{trace } Q$.

$$\begin{bmatrix} He\{\tilde{P}_2 \tilde{A}\} & \tilde{P}_2 \tilde{B}_0 \\ * & -I \end{bmatrix} < O \quad (3.73)$$

$$\begin{bmatrix} Q & C_0 + D_{02} K C_2 \\ * & P_2 \end{bmatrix} > O \quad (3.74)$$

$$\begin{bmatrix} He\{\tilde{P}_\infty \tilde{A}\} & \tilde{P}_\infty \tilde{B}_1 & * \\ * & -\gamma_\infty I & * \\ \tilde{C}_1 & \tilde{D}_{11} & -\gamma_\infty \end{bmatrix} < O \quad (3.75)$$

$$P_\infty > O \quad (3.76)$$

$$\tilde{P}_2 = \begin{bmatrix} P_2 & G_{21} \\ O & G_{22} \end{bmatrix}, \quad \tilde{P}_\infty = \begin{bmatrix} P_\infty & G_{\infty 1} \\ O & G_{\infty 2} \end{bmatrix} \quad (3.77)$$

Solusi dari problem optimasi yaitu didapatkan nilai *gain observer*.

3.4 Uji Performansi PFTC

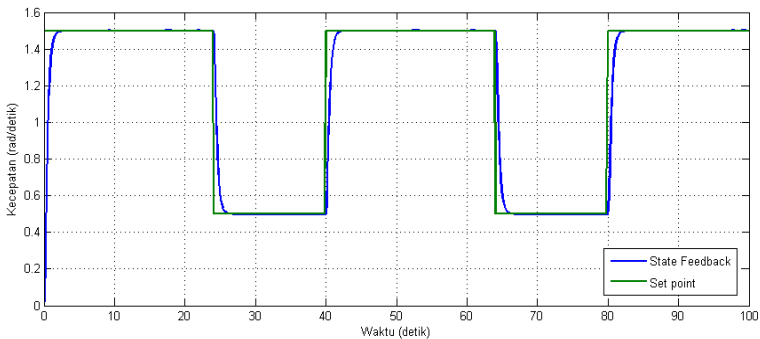
Setelah dilakukan integrasi atau penggabungan *observer* dengan sistem kontrol (PFTC) diperlukan uji performansi PFTC. Pada sistem PFTC besarnya kesalahan sudah ditentukan terlebih dahulu saat tahap perancangan sistem. Besarnya kesalahan pada saat tahap perancangan diberi simbol yaitu F_s . Kemudian dilakukan uji performansi berupa uji kesalahan bias, uji kesalahan sensitifitas dan uji ketidakpastian. Besar kesalahan diberikan pada setiap uji pada *simulink* untuk setiap nilai F_s yang diberikan. Sehingga dapat dicari pada nilai F_s berapa besar kesalahan sebesar apapun masih bisa ditangani dan menghasilkan respon yang paling bagus. Serta dapat diambil kesimpulan dari uji tersebut tentang pengaruh penetapan nilai besar kesalahan atau F_s terhadap setiap uji yang dilakukan.

BAB IV

ANALISIS DATA DAN PEMBAHASAN

4.1 Uji *Tracking* Kontroler

Nilai *gain* kontroler yang didapatkan dari persamaan 3.12 diperlukan pengujian untuk mengetahui respon yang dihasilkan dari kontroler. Pengujian kontroler untuk mengetahui respon dari *gain* kontroler apakah bisa *tracking* saat ada perubahan *set point*. Berikut respon dari uji *tracking* kontrollor :



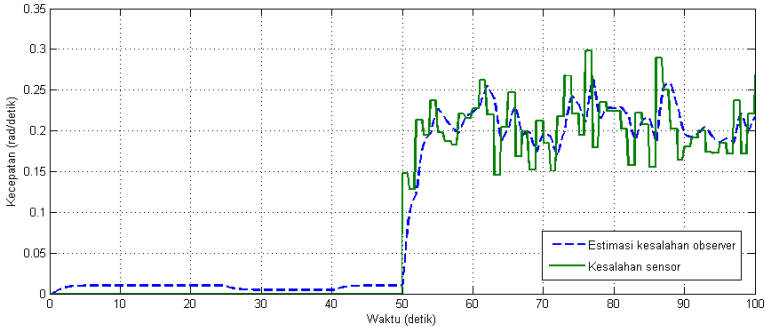
Gambar 4. 1 Respon uji kontroler

Gambar 4.1 menunjukkan dua grafik yaitu grafik berwarna hijau yang menunjukkan nilai *set point* dan grafik berwarna biru menunjukkan hasil respon kontroler. Dari grafik terlihat bahwa respon *state feedback* dapat *tracking* ketika *set point* sebesar 1,5 berubah ke *set point* sebesar 0,5. Hal ini membuktikan bahwa diketahui bahwa kontroler yang dirancang dapat bekerja dengan baik karena mampu mencapai mengikuti nilai *set point* . Hasil respon menunjukkan karakteristik respon nilai *maximum overshoot* ataupun *maximum undershoot* dan *error steady state* sebesar 0% sedangkan nilai *settling time* sebesar 0,60 detik.

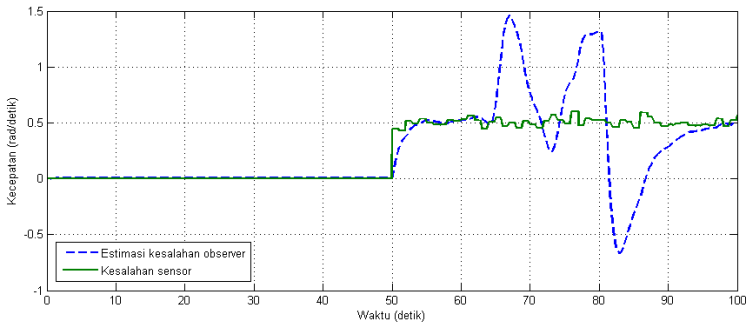
4.2 Uji Hasil Estimasi *Observer*

Pengujian ini dilakukan untuk menganalisa *observer* yang sudah dirancang apakah mampu mengestimasi kesalahan sensor. Uji estimasi *observer* ini dilakukan dengan sistem tanpa diberikan

kesalahan dan sistem dengan pemberian kesalahan sebesar 50 %,sehingga didapatkan respon sebagai berikut :



Gambar 4. 2 Hasil perbandingan kesalahan aktual dengan estimasi kesalahan sensor tanpa kesalahan



Gambar 4. 3 Hasil perbandingan kesalahan aktual dengan estimasi kesalahan sensor dengan kesalahan

Pada gambar 4.2 dan 4.3 dapat dilihat bahwa garis hijau menunjukkan kesalahan aktual dan garis biru menunjukkan estimasi kesalahan *observer*. Pada sistem tanpa kesalahan maupun dengan pemberian kesalahan sebesar 50%, diketahui bahwa hasil estimasi kesalahan sensor mengikuti kesalahan actual. Sehingga dapat disimpulkan bahwa *observer* sudah bekerja dengan benar untuk mengestimasi besarnya kesalahan.

4.3 Uji Performansi Kesalahan Sensor

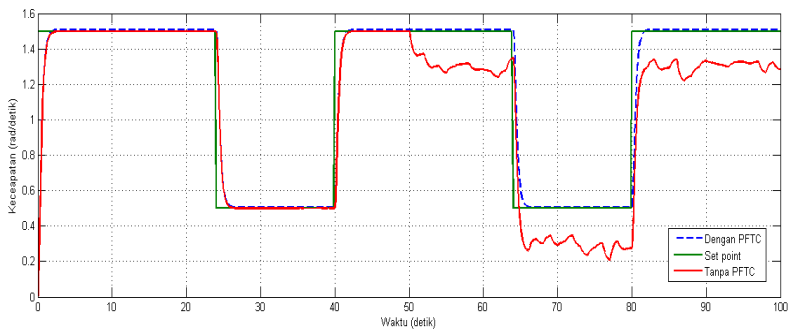
Uji performansi dilakukan untuk mengetahui algoritma *passive fault tolerant control* (PFTC) yang telah dibuat dapat bekerja dengan baik atau tidak ketika sistem diberikan kesalahan. Pengujian dilakukan dengan memberikan kesalahan berupa kesalahan bias dan kesalahan sensitifitas pada sensor.

4.3.1 Uji Kesalahan Bias Sensor

Pada PFTC besarnya kesalahan sudah ditentukan terlebih dahulu pada proses perancangan dengan memberikan nilai ketetapan kesalahan sensor yang disimbolkan dengan F_s . Kemudian ketika tahap simulasi di *Simulink* dirubah menjadi f_s yang menunjukkan besar kesalahan pada sensor untuk tahap simulasi. Uji performansi ini dilakukan dengan cara mencari nilai dari F_s supaya berapa pun besar kesalahan yang diberikan masih bisa ditangani dan memberikan respon yang paling bagus ketika kesalahan paling besar. Dengan memberikan kesalahan bias dan kesalahan sensitivitas maka dapat diketahui pengaruh pemberian kesalahan tersebut pada penentuan F_s .

Untuk mencari nilai F_s maka dilakukan pengujian dengan memberi nilai F_s sebesar 0,3 dengan kesalahan bias sebesar 13%, 47% , 60%, dan 100%. Sehingga didapatkan respon sebagai berikut :

- a. Nilai $F_s = 0,3$ dengan kesalahan bias 13%



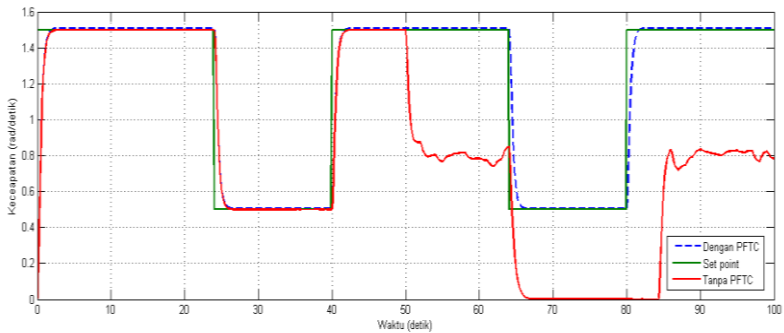
Gambar 4. 4 Respon dengan $F_s = 0,3$ kesalahan bias 13%

Tabel 4. 1 Performansi sistem $F_s = 0.3$ kesalahan bias 13%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Maximum Undershoot</i>	0%	0%	18%	60%
<i>Error Steady State</i>	0,09%	0,09%	13.,%	30%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.4 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan bias pada detik ke 50, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *maximum undershoot* yang lebih kecil yaitu 0% untuk sistem dengan PFTC dan 18% untuk sistem tanpa PFTC. *Error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 13,3% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0,5 atau saat respon turun, sistem dengan PFTC tidak memiliki *maximum undershoot* sedangkan sistem tanpa PFTC sebesar 30%. *Error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0.09% untuk sistem dengan PFTC dan 30% sistem tanpa PFTC. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem. Nilai yang dihasilkan dari hasil estimasi *observer* menunjukkan nilai yang benar yaitu sama dengan besarnya kesalahan sebesar 0,2 volt. Begitu juga dengan nilai respon output menunjukkan nilai yang sama pula yaitu 0,2 volt.

b. Nilai $F_s = 0,3$ dengan kesalahan bias 47%



Gambar 4. 5 Respon dengan $F_s = 0,3$ kesalahan bias 47%

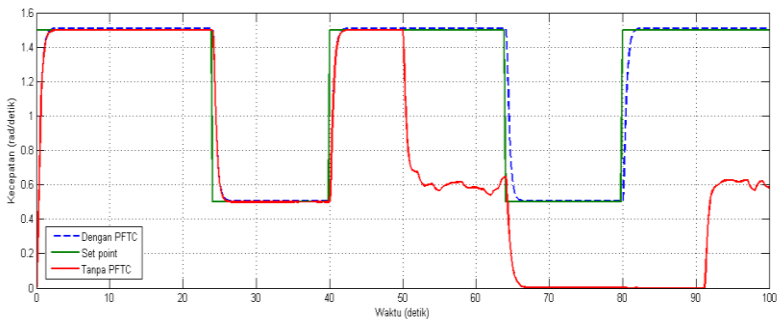
Tabel 4. 2 Performansi sistem $F_s = 0,3$ kesalahan bias 47%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Maximum Undershoot</i>	0%	0%	50%	0%
<i>Error Steady State</i>	0,09%	0,09%	46%	50%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.5 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan bias pada detik ke 50, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *maximum undershoot* yang lebih kecil yaitu 0% untuk sistem dengan PFTC dan 50% untuk sistem tanpa PFTC. *Error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 46% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0,5 atau saat respon turun *error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0.09% untuk sistem dengan PFTC dan 50%

sistem tanpa PFTC. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem. Nilai yang dihasilkan dari hasil estimasi *observer* menunjukkan nilai yang salah yaitu 0,59, dimana nilai tersebut tidak sama dengan besarnya kesalahan yakni 0,7 volt. Namun nilai respon output menunjukkan nilai yang sama dengan besarnya kesalahan yaitu 0,7 volt.

c. Nilai $F_s = 0,3$ dengan kesalahan bias 60%



Gambar 4. 6 Respon dengan $F_s = 0,3$ kesalahan bias 60%

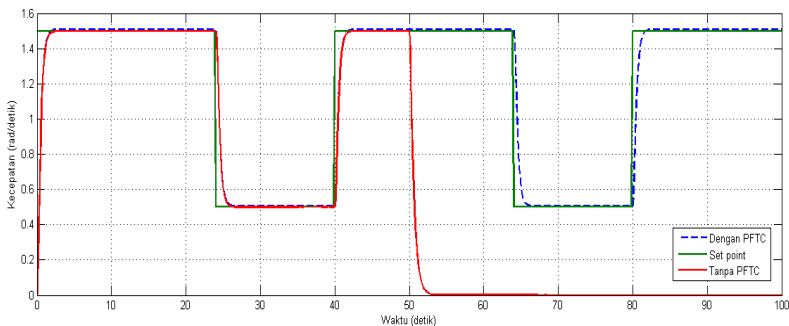
Tabel 4. 3 Performansi sistem $F_s = 0,3$ kesalahan bias 60%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Maximum Undershoot</i>	0%	0%	63%	0%
<i>Error Steady State</i>	0,09%	0,09%	58%	50%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.6 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan bias pada detik ke 50, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *maximum undershoot* yang lebih kecil yaitu 0% untuk sistem dengan PFTC dan 63%

untuk sistem tanpa PFTC. *Error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 58% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0,5 atau saat respon turun *error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0.09% untuk sistem dengan PFTC dan 50% sistem tanpa PFTC. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem. Nilai yang dihasilkan dari hasil estimasi *observer* menunjukkan nilai yang salah yaitu -0,06, dimana nilai tersebut tidak sama dengan besarnya kesalahan yakni 0,9 volt. Namun nilai respon output menunjukkan nilai yang sama dengan besarnya kesalahan yaitu 0,9 volt.

d. Nilai $F_s = 0,3$ dengan kesalahan bias 100%



Gambar 4. 7 Respon dengan $F_s = 0,3$ kesalahan bias 100%

Tabel 4. 4 Performansi sistem $F_s = 0,3$ kesalahan bias 100%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Error Steady State</i>	0,09%	0,09%	-	-
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.7 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan bias pada detik ke 50,

respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *error steady state* dan *settling time* yang lebih kecil dari sistem tanpa PFTC. Hal tersebut disebabkan karena dengan memberikan kesalahan sebesar 100% respon sistem tanpa PFTC tidak menghasilkan sinyal. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem meskipun diberikan besaran kesalahan sebesar 100% atau 1,5 volt.

Secara keseluruhan hasil respon sistem ketika kesalahan bias terjadi sebesar 13%, 47%, 60% dan 100% sistem dengan PFTC memiliki nilai *error steady state* yang jauh lebih kecil dibandingkan sistem tanpa PFTC baik saat *set point* 1,5 dan 0,5. Ketika diberi kesalahan bias pada detik ke 50, respon saat *set point* 1,5 atau saat naik memiliki nilai *error steady state* sebesar 13% untuk kesalahan bias 13%, 46% untuk kesalahan bias 47%, 58% untuk kesalahan bias 60% dan tidak stabil untuk kesalahan bias 100%. Sedangkan untuk sistem dengan PFTC memiliki *error steady state* sebesar 0,09% untuk semua kesalahan bias yang terjadi. Begitu pula dengan nilai *maximum undershoot* yang lebih kecil dengan sistem dengan PFTC dibanding sistem tanpa PFTC. Semakin besar kesalahan bias yang diberikan maka *maximum undershoot* dan *error steady state* sistem semakin besar. Sistem dengan PFTC mampu mencapai stabil dengan sangat cepat karena sistem ini *kontinyu*, sehingga apabila diberi kesalahan sistem PFTC dapat kembali ke keadaan stabil dengan cepat.

Dengan memberikan nilai tetapan besaran kesalahan atau F_s sebesar 0,3 volt, berpengaruh terhadap dinamika *observer*. Dimana ketika kesalahan yang diberikan saat simulasi dibawah nilai F_s tersebut sebesar 0,2 volt (13%), maka hasil estimasi *observer* menunjukkan nilai yang benar sesuai nilai kesalahan. Respon output juga akan menunjukkan nilai yang sama benarnya. Namun apabila besarnya kesalahan sebesar 0,7 volt (47%), 0,9 volt (60%) dan 1,5 volt (100%) diatas waktu training atau $F_s=0,3$, maka hasil estimasi *observer* salah tetapi nilai respon output menunjukkan nilai yang benar. Sehingga dapat disimpulkan bahwa nilai F_{gs} berpengaruh terhadap *observer*. Sistem dengan

kompensasi *observer* memiliki respon yang tetap bagus walaupun diatas maupun dibawah nilai training. Sedangkan sistem tanpa kompensasi *observer* dipengaruhi oleh nilai saat *training*, apabila dibawah nilai *training* bisa menunjukan nilai yang benar dan respon yang benar sedangkan diatas *training* akan menunjukan nilai dan respon yang salah.

Dengan nilai $F_s = 0,3$ besar kesalahan bias terbesar yang dapat ditangani yaitu 100%. Hal ini disebabkan sensor yang digunakan dalam sistem ini menggunakan *soft* sensor sehingga saat diberi kesalahan 100% sensor menggantikan motor DC. *Soft* sensor dapat bekerja didasarkan pada *state observer* yang telah dibuat sehingga dapat mengestimasi besarnya kesalahan yang terjadi pada sistem. Selain menggunakan nilai besar kesalahan sebesar 0,3 dilakukan juga pengujian dengan menggunakan besar kesalahan sebesar 0,1, 0,5 dan 0,8 untuk mengetahui pada F_s berapa kesalahan bias terbesar masih dapat ditangani. Berikut merupakan perbandingan hasil respon dari nilai f_s :

Tabel 4. 5 Perbandingan Hasil Respon Nilai F_s

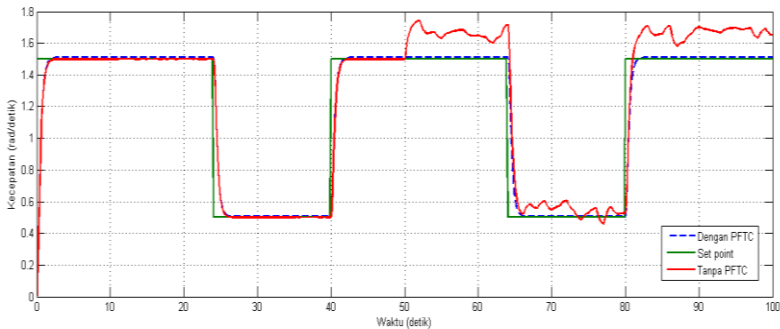
Nilai F_s	<i>Error steady state</i>	<i>Settling time</i>
0,3	0,09%	0,6 s
0,1	0,2%	51 s
0,5	0,4%	51.5 s
0,8	1,5%	53.4 s

Dengan nilai F_s sebesar 0,3, 0,1, 0,5 dan 0,8 dengan besar kesalahan bias 100% (1,5 volt) respon yang dihasilkan sama namun apabila diperbesar respon sistem dengan PFTC memiliki respon yang berbeda beda. Dengan nilai f_s sebesar 0,3, sistem dengan PFTC memiliki *error steady state* sebesar 0,09%, 0,2% untuk f_s 0,1, 0,4% untuk f_s 0,5 dan 1,5% untuk f_s 0,8. Nilai *settling time* dari f_s 0,3 juga lebih kecil daripada f_s 0,1, 0,5 dan 0,8. Sehingga nilai f_s sebesar 0,3 merupakan nilai yang paling tepat yang dapat mengatasi kesalahan terbesar sebesar 100% dengan hasil respon yang bagus dan nilai sistem dengan PFTC maupun sistem tanpa PFTC menunjukan nilai yang benar.

4.3.2 Uji Kesalahan Sensitifitas Sensor

Selain uji kesalahan bias, diperlukan juga uji sensitifitas untuk mengetahui respon sistem dengan menggunakan PFTC dan tanpa menggunakan PFTC. Serta untuk mencari nilai F_s yang masih mampu menangani kesalahan sensitifitas yang paling besar. Untuk mencari nilai F_s maka dilakukan pengujian dengan memberi nilai F_s sebesar 0,3 dengan kesalahan sensitifitas sebesar 10%, 50%, 90% dan 100%. Sehingga didapatkan respon sebagai berikut :

- a. Nilai $F_s = 0,3$ dengan kesalahan sensitifitas 10%



Gambar 4. 8 Respon dengan $F_s = 0,3$ kesalahan sensitifitas 10%

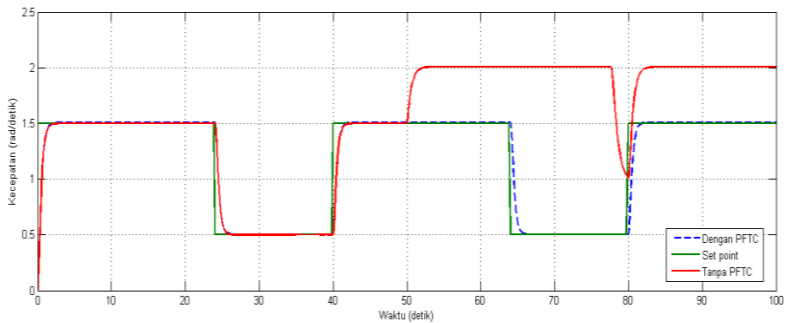
Tabel 4. 6 Performansi sistem $F_s = 0,3$ kesalahan sensitifitas 10%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Maximum Overshoot</i>	0%	0%	18%	20%
<i>Error Steady State</i>	0,09%	0,09%	20%	20%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.8 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan sensitifitas pada detik ke 50, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan

PFTC memiliki respon yang lebih baik karena nilai *maximum overshoot* yang lebih kecil yaitu 0% untuk sistem dengan PFTC dan 18% untuk sistem tanpa PFTC. *Error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 20% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0.5 atau saat respon turun *error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 20% sistem tanpa PFTC. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem.

b. Nilai $F_s = 0,3$ dengan kesalahan sensitifitas 50%



Gambar 4. 9 Respon dengan $F_s = 0,3$ kesalahan sensitifitas 50%

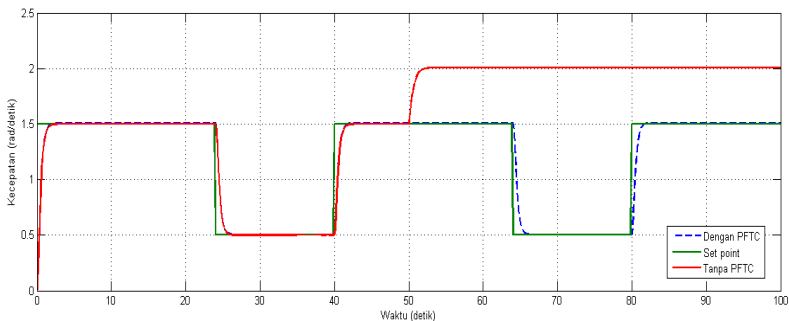
Tabel 4. 7 Performansi sistem $F_s = 0,3$ kesalahan sensitifitas 50%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Error Steady State</i>	0,09%	0 %	50,2%	0%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.9 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan sensitifitas pada detik ke

50, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 50% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0,5 atau saat respon turun respon tanpa sistem PFTC tidak dapat mengikuti *set point* namun dengan PFTC respon dapat kembali mencapai *steady*. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem.

c. Nilai $F_{ss} = 0,3$ dengan kesalahan sensitifitas 90%



Gambar 4. 10 Respon dengan $F_s = 0,3$ kesalahan sensitifitas 90%

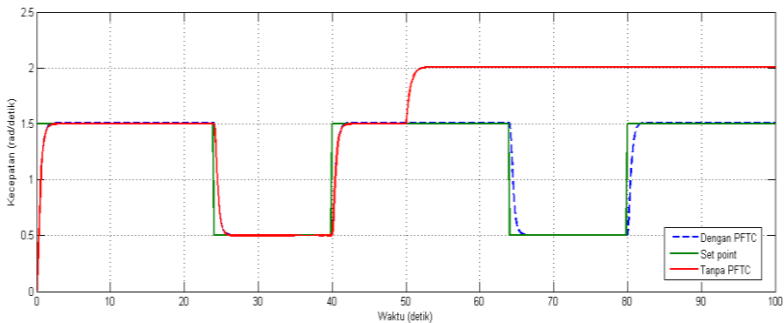
Tabel 4. 8 Performansi sistem $F_s = 0,3$ kesalahan sensitifitas 90%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Error Steady State</i>	0,09%	0 %	50,7%	0%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.10 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan sensitifitas pada detik ke

50, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 50% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0,5 atau saat respon turun respon tanpa sistem PFTC tidak dapat mengikuti *set point* namun dengan PFTC respon dapat kembali mencapai *steady*. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem.

d. Nilai $F_s = 0,3$ dengan kesalahan sensitifitas 100%



Gambar 4. 11 Respon dengan $F_s = 0,3$ kesalahan sensitifitas 100%

Tabel 4. 9 Performansi sistem $F_s = 0,3$ kesalahan sensitifitas 100%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Error Steady State</i>	0,09%	0 %	52%	0%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.11 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi kesalahan sensitifitas pada detik ke

50, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 50% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0,5 atau saat respon turun respon tanpa sistem PFTC tidak dapat mengikuti *set point* namun dengan PFTC respon dapat kembali mencapai *steady*. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem walaupun diberi kesalahan sampai 100%.

Secara keseluruhan hasil respon sistem ketika kesalahan sensitifitas terjadi sebesar 10%, 50%, 90%, dan 100% sistem dengan PFTC memiliki nilai *error steady state* yang jauh lebih kecil dibandingkan sistem tanpa PFTC. Sistem tanpa PFTC tidak dapat kembali mencapai *set point* 1,5 dan 0,5 dengan nilai *error steady state* sebesar 20% untuk kesalahan sensitifitas 10%, 50,2 % untuk kesalahan sensitifitas 50%, 50,7% untuk kesalahan sensitifitas 90% dan 52% untuk kesalahan sensitifitas 100% . Semakin besar kesalahan sensitifitas yang mendekati nilai stabil yaitu bernilai 1, maka hasil respon berupa nilai *error steady state* semakin kecil karena mendekati nilai yang stabil. Sedangkan untuk sistem dengan PFTC memiliki *error steady state* sebesar 0,09% untuk semua kesalahan sensitifitas yang terjadi. Sistem dengan PFTC mampu mencapai stabil dengan sangat cepat karena sistem ini kontinyu sehingga apabila diberi kesalahan , sistem PFTC dapat kembali ke keadaan stabil dengan cepat.

Dengan memberikan nilai $F_s = 0,3$ untuk kesalahan bias sebesar apapun berpengaruh terhadap *observer*. Namun untuk kesalahan sensitifitas ini tidak dipengaruhi nilai *training* seperti pada kesalahan bias. Walaupun diberikan nilai kesalahan diatas atau dibawah nilai *training* yaitu 0,3, nilai hasil estimasi *observer* dan nilai respon output yang dihasilkan tidak sama dengan besar kesalahan yang diberikan. Tetapi *observer* tetap bekerja baik karena respon yang dihasilkan dengan sistem PFTC dapat menangani kesalahan yang ada. Dengan nilai $F_s=0,3$ dapat menangani kesalahan sensitifitas sampai sebesar 100%. Hal ini

sama dengan pengujian kesalahan bias dimana besar kesalahan sampai 100% pun masih bisa ditangani. Karena sensor yang digunakan dalam sistem ini adalah *soft* sensor maka besar kesalahan sampai 100% pun dapat diatasi dan menghasilkan respon yang kembali stabil.

Selain menggunakan besar kesalahan sebesar 0,3, dilakukan pengujian lagi dengan menggunakan variasi besar kesalahan sebesar 0,1, 0,5 dan 0,8 untuk mengetahui kesalahan sensitifitas yang paling besar yang masih dapat ditangani oleh sistem. Berikut merupakan perbandingan hasil respon dari nilai f_s :

Tabel 4. 10 Perbandingan Hasil Respon Nilai F_s

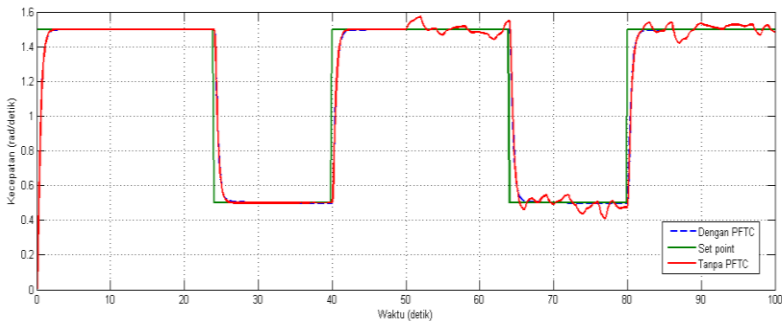
Nilai F_s	<i>Error steady state</i>	<i>Settling time</i>
0,3	0,09%	0,6 s
0,1	0,5%	52 s
0,5	0,4%	51,5 s
0,8	1,5%	51 s

Secara keseluruhan dengan nilai F_s sebesar 0,1, 0,3, 0,5, dan 0,8 dengan nilai kesalahan sensitifitas 100%, respon yang dihasilkan hampir sama semua. Namun terdapat hal yang membedakan dari keempat respon tersebut yaitu *error steady state* dan *settling time* saat sistem dengan PFTC memperbaiki kesalahan pada detik ke50. Dengan f_s sebesar 0,1 masih terdapat *error steady state* 0,5%, 0,09% untuk f_s 0,3, 0,4% untuk f_s 0,5 dan 1,5% untuk f_s 0,8. Dengan nilai f_s sebesar 0,3 memiliki *error steady state* yang lebih kecil daripada nilai f_s lain. Selain itu *settling time* juga lebih kecil daripada f_s lain. Sehingga nilai $F_s = 0,3$ adalah nilai yang paling tepat karena dapat menangani kesalahan terbesar sebesar 100% dan menghasilkan respon yang baik.

4.3.3 Uji Ketidakpastian

Uji ketidakpastian ini dilakukan untuk mengetahui apakah sistem dengan PFTC mampu mengatasi besar ketidakpastian yang diberikan pada tahanan kumparan jangkar (R). Dengan nilai besar

kesalahan atau F_s sebesar 0,3 sistem diberi ketidakpastian sebesar 10%(0,1), sehingga menghasilkan respon sebagai berikut :



Gambar 4. 12 Respon dengan ketidakpastian 10%

Tabel 4. 11 Performansi dengan ketidakpastian 10%

Parameter	Dengan PFTC		Tanpa PFTC	
	Naik	Turun	Naik	Turun
<i>Maximum Overshoot</i>	0%	0%	6%	16%
<i>Maximum Undershoot</i>	0%	0%	3%	20%
<i>Error Steady State</i>	0,09%	0,09%	5%	5%
<i>Settling Time</i>	0,60 s	0,06 s	-	-

Pada gambar 4.12 respon sistem tanpa PFTC ditunjukkan oleh grafik berwarna *orange*, sistem dengan PFTC ditunjukkan grafik berwarna biru dan *set point* ditunjukkan dengan grafik berwarna hijau. Ketika diberi ketidakpastian, respon saat *set point* 1,5 atau saat naik menunjukkan bahwa sistem dengan PFTC memiliki respon yang lebih baik karena nilai *maximum overshoot* dan *undershoot* yang lebih kecil yaitu 0% untuk sistem dengan PFTC sedangkan 6% dan 3% untuk sistem tanpa PFTC. *Error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 5% sistem tanpa PFTC. Begitu juga saat *set point* sebesar 0,5 atau

saat respon turun, sistem dengan PFTC tidak memiliki *maximum overshoot* dan *undershoot* sedangkan sistem tanpa PFTC sebesar 16% dan 20%. *Error steady state* sistem dengan PFTC juga lebih kecil dibanding sistem tanpa PFTC yaitu sebesar 0,09% untuk sistem dengan PFTC dan 5% sistem tanpa PFTC. Hal ini membuktikan bahwa sistem PFTC dapat meningkatkan performansi dari suatu sistem ketika diberikan ketidakpastian dalam pemodelan motor DC yaitu pada tahanan kumparan (R).

“Halaman ini sengaja dikosongkan”

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis data yang telah dilakukan didapatkan kesimpulan dari tugas akhir mengenai perancangan *Passive Fault Tolerant Control* (PFTC) pada pengendalian kecepatan sistem motor DC dengan kesalahan pada sensor adalah sebagai berikut :

- Telah dilakukan perancangan algoritma PFTC berbasis *observer* untuk mengestimasi kesalahan yang terjadi pada sensor dan mampu menjaga respon kecepatan untuk tetap pada *set point*. Parameter ketetapan kesalahan yang terbaik adalah 0,3.
- Sistem kontrol *state feedback* yang dirancang mampu *tracking set point* dengan nilai K sebesar 15,49 dan 5,50 serta nilai K_i sebesar 151,50.
- Sistem dengan algoritma PFTC dapat mengestimasi kesalahan bias sebesar 13%, 47%, 60% dan 100% . Kesalahan bias terbesar yang masih mampu diatasi sebesar 60%.
- Sistem dengan algoritma PFTC dapat mengestimasi kesalahan sensitifitas 10%, 50%, 90% dan 100% . Kesalahan sensitifitas terbesar yang masih mampu diatasi sebesar 90%.
- Hasil respon sistem dengan PFTC ketika diberi kesalahan bias 60% saat respon naik tidak menghasilkan *maximum undershoot* sedangkan sistem tanpa PFTC menghasilkan *maximum undershoot* sebesar 63%. Respon sistem dengan PFTC tidak menghasilkan *error steady state* sedangkan sistem tanpa PFTC menghasilkan *error* sebesar 58%.
- Hasil respon sistem dengan PFTC ketika diberi kesalahan sensitifitas 90% saat respon naik tidak menghasilkan *error steady state* namun menghasilkan *settling time* sebesar 0,60 s. Respon sistem tanpa PFTC menghasilkan *error steady state* sebesar 50.7% dan tidak dapat mencapai nilai *set point*.
- PFTC mampu mengatasi ketidakpastian terbesar sebesar 10% yang terjadi pada tahanan kumparan (R).

5.2 Saran

Dari tugas akhir ini terdapat saran yang dapat digunakan untuk penelitian selanjutnya, yakni pengembangan algoritma PFTC dapat dilakukan secara *real time* pada *miniplant* dengan berbagai macam penggunaan strategi kontrol .

DAFTAR PUSTAKA

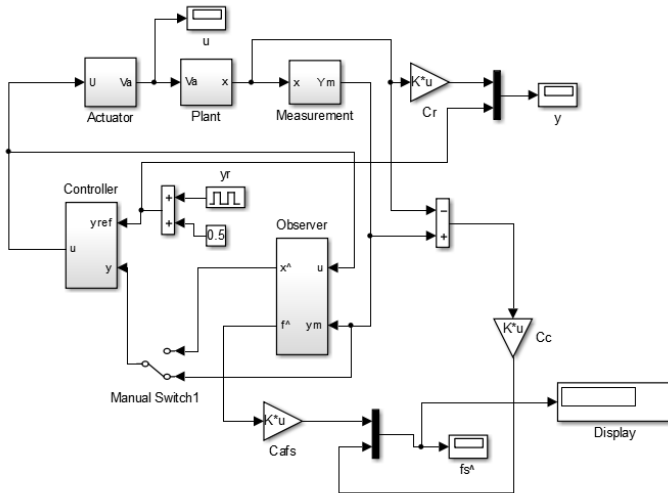
- Alwi, H., & Edwards, C. (2011). *Fault Detection and Fault-Tolerant Control using Sliding Modes*. New York: Springer-Verlag.
- Basilio, J. C. (2002). *Design of PI and PID Controlers With Transient Performance Specification*. *IEEE Transactions On Education*, Vol. 45, No.4.
- Blanke M, S. (2001). *Concepts and methods in fault tolerant control*. *Proc.American Control Conf*. Arlington.
- Burridge, M. J., & Qu, Z. (2003). *An improved nonlinear control design for series DC motors*. *Computers and Electrical Engineering*, vol. 29, 730-735.
- Campos-Delgado. (2008). *Fault-tolerant control in variable speed drives: a survey*. *IET Electrical Power Application*, vol. 2 (no. 2), 121-134.
- Chen, G. (2004). *System analysis using redundancy of descriptor representation*. *IEEE*, 231-236.
- Delgado, D. C. (2007). *Fault-tolerant control in variable speed drives: a survey*. *IET Electric Power Applications*.
- Erwin Susanto, P. (2015). *Pengantar Kontrol Maju*. Telkom University.
- G.Rigatos, G. (2007). *Adaptive fuzzy control of DC motors using state and output feedback*. *IEEE*.
- Indriawati, K. (2015). *Robust Fuzzy Observer-Based Fault Tolerant Tracking Control for Nonlinear Systems with Simultaneous Actuator and Sensor Faults: Application to a DC Series Motor Speed Drive*. *International Review of Automatic Control (I.R.E.A.CO.)*, Vol. 8, N. 6, 375-385.
- Indriawati, K., Agustinah, T., & Jazidie, A. (2015). *Robust observer-based fault tolerant tracking control for linear systems with simultaneous actuator and sensor faults: Application to a DC motor system*. *International Review on Modelling and Simulations (IREMOS)*, 8 (4), 410-417.

- John Doyle, B. F. (1990). *Feedback Control Theory*. Macmillan Publisihing.
- Katherin I, T. A. (2015). *Robust Observer-Based Fault Tolerant Tracking Control for Linier Systems with Simultanes Actuator and Sensor Faults:Application to a DC Motor System. International Review on Modelling Simulations*, Vol 8 N 4.
- Kemin Zhou, J. C. (1996). *Robust and Optimal Control*. Prentice Hall.
- Koksal, M. (2007). *Position Control of a Permanent Magnet DC Motor by Model Reference Adaptive Control* . *IEEE*, 112-117.
- Kuo, B. (1995). *Teknik Kontrol Automatik*. Jogjakarta: Aditya Media.
- Noura, H., Theilliol, D., Ponsart, J. C., & Chamseddine, A. (2009). *Fault-tolerant Control Systems: Design and Practical Applications*. London: Springer-Verlag.
- Ogata, K. (1997). *Modern Control Engineering 3rd ed*. London: Prentice Hall International.
- Oliveira, V. (2006). *Robust Conroller Enhanced With Design and Implementasi*. *IEEE*, 3.
- P.C, K. (2002). *Analysis of electric machinery and drive systems*. *IEEE Press*.
- Scherer, C. (1997). *Multiobjective Output-Feedback Control via LMI Optimization*. *IEEE*.
- Sebe, N. (2007). *A New Dilated LMI Characterization and Iterative Control System Synthesis*.
- Setyaningrum, D. (2012). *Desain dan Implementasi Model Reference Adaptive Control untuk Pengaturan Tracking Optimal Posisi Motor DC*. *JURNAL TEKNIK POMITS Vol. 1, No. 1*, 1-6.
- Tsui, C.-c. (2004). *Robust Control System Design*. Marcel Dekker. Inc, New York .
- Wai, P. A. (2007). *Analysis on Modeling and Simulink of DC motor and its Driving System Used for WheeledMobile Robot*. *IEEE*, 25-33.

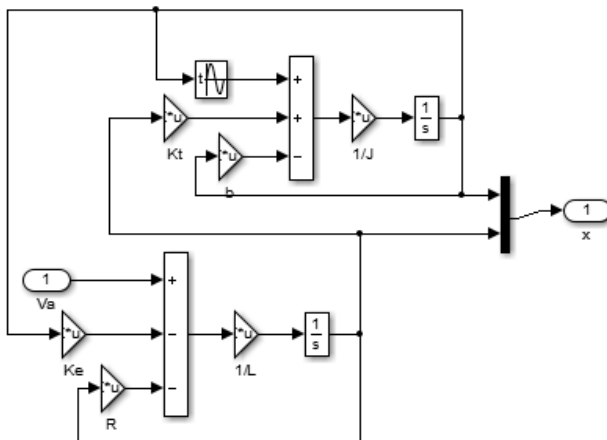
- Zhang, Y., & Jiang, J. (2008). *Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems. Annual Reviews in Control* vol 32 (no.2), 229-252.
- Zhao, Q. (1999). *Fault Tolerant Control System Design*.

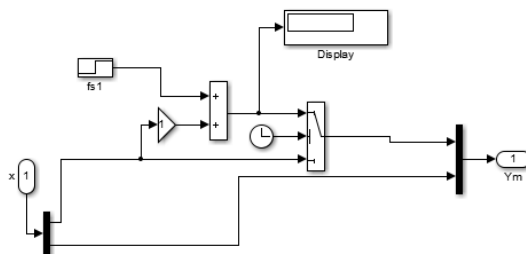
LAMPIRAN A SUBSYSTEM *SIMULINK* MATLAB R2013a

a. General Design Pengendalian Kecepatan Motor DC



b. Plant Motor DC





LAMPIRAN B
LISTING PROGRAM MATLABR2013a m.file
SISTEM MOTOR DC

```
J = 0.01;
b = 0.1;
Ke = 0.01;
R = 1;
Lin = 0.5;

A = [-b/J    Ke/J
     -Ke/Lin  -R/Lin];
B = [0
     1/Lin];
C = [1    0; 0    1];
Cr = [1    0];
D = 0;

nx=size(A,1);
ny=size(C,1);
nu=size(B,2);
nr=1;          % number of reference inputs

% design PI controller
Ac = [A, zeros(nx,nr); C(1:nr,:) , zeros(nr) ];
Bc = [B; 0];
p1 = -10 + i;
p2 = -10 - i;
p3 = -3;
Kc = place(Ac,Bc,[p1 p2 p3]);
Kp = Kc(1:2);
Ki = Kc(3);

Dd = [-1 0; 0 -0.2];
Dw = [1; 0];
F = [1; 0];
E = B;

nw = size(Dw,2);
```

```

nd = size(Dd,2);
nfs = size(F,2);
nfa = size(E,2);

Aa = [A zeros(nx,ny); C -eye(ny)];
Ba = [B; zeros(ny,nu)];
Ea = [E zeros(nx,nfs); zeros(ny,nfa) F];
%Fa = [zeros(nx,nw); Dw];
%Fa = [Dd;zeros(ny,nd)];
Fa = [Dd zeros(nx,nw);zeros(ny,nd) Dw];
Ca = [zeros(ny,nx) eye(ny)];
Cao = [C zeros(ny)];
Cafa = [ones(1,nfa) zeros(1,nfs)];
%Bo = [zeros(n+p,fa+fs); eye(fa+fs)];
Bo = [Fa zeros(nx+ny,nfa+nfs);
zeros(nfa+nfs,nd+nw) eye(nfa+nfs)];

%%% FTC against actuator and/or sensor failure
Aas=[Aa Ea; zeros(nfa+nfs,nx+ny)
zeros(nfa+nfs)];
Cas=[Ca zeros(ny,nfa+nfs)];

R4=eye(ny);
Q4=eye(nx+ny+nr*2)*1e2;
L4=lqr(Aas',Cas',Q4,R4);

L=L4';

Lqr=L4';

addpath inputs101

ws1=tf([0.72,0.3],[1,1e-2]);
ws2=tf([0.8,1],[1,1e-2]); % weight function for
tracking performance
nn=2; % number of generalized
plants
ps=struct('sys',cell(1,nn),'spec','hinf','Fopt',
1,'bound',0);

```



```

% generalized plant (normal case)
Fga = 1; Fgs = 1; Fga1 = 0; Fgr = 1;
[ta,tb,tc,td]=linmod2('clsys');
ps(1).sys=ss(ta,tb,tc,td);

% generalized plant (sensor failure case)
Fga = 1; Fgs = 0.5; Fga1 = 0; Fgr = 1;
[ta,tb,tc,td]=linmod2('clsys');
ps(2).sys=ss(ta,tb,tc,td);

lcmax=100; % maxum iteration
L_init=ss([],[],[],Lqr);
[gg,ga,Lopt]=multiobjective101(ps,L_init,[],100)
;

gg
Lopt

L=Lopt.d;

```

LAMPIRAN C

$$A = \begin{bmatrix} -10.0000 & 1.0000 \\ -0.0200 & -2.0000 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$C_r = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$D = 0$$

$$A_C = \begin{bmatrix} -10 & 1 & 0 \\ -0.02 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$B_C = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

$$A_a = \begin{bmatrix} -10 & 1 & 0 & 0 \\ -0.02 & -2 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$B_a = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}$$

$$E_a = \begin{bmatrix} 0 & 0 \\ 2 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$C_a = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$ta = \begin{bmatrix} -10.0000 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0200 & -2 & 0 & -449.9800 & -38.0000 & 0 & 0 & 0 & 0 & -2 & 0 & -5 \times 10^3 \\ 0 & 0 & -0.0100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -450.0000 & -40.0000 & 0 & 0 & 0 & 0 & -2 & 0 & -5 \times 10^3 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0.5000 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$tb = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$tc = \begin{bmatrix} -0.7200 & 0 & 0.5856 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$td = \begin{bmatrix} 0.7200 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

BIODATA PENULIS



Nama penulis Amalia Zabda Fasya. Penulis dilahirkan di Kediri, tanggal 1 Agustus 1996 dari ayah adalah Siswanto dan ibu adalah Niswaturosidah. Saat ini penulis tinggal di Jalan Ngadiluwih Wates Purworejo Kandat, Kediri, Provinsi Jawa Timur. Penulis telah menyelesaikan pendidikan tingkat dasar di MI Al-Fajar Kandat Kediri pada tahun 2008, pendidikan menengah pertama di SMPN 1 Ngadiluwih Kediri pada tahun 2011, pendidikan menengah atas di SMAN 1 Kediri pada tahun 2014, dan sedang menempuh pendidikan S1 Teknik Fisika FTI di Institut Teknologi Sepuluh Nopember Surabaya hingga sekarang.

Pada bulan Maret 2018 penulis telah menyelesaikan Tugas Akhir dengan judul **Perancangan *Passive Fault Tolerant Control* (PFTC) pada Pengendalian Kecepatan Sistem Motor DC dengan Kesalahan pada Sensor**. Bagi pembaca yang memiliki saran, kritik, atau ingin berdiskusi lebih lanjut mengenai tugas akhir ini maka dapat menghubungi penulis melalui *email*: amaliazabda02@gmail.com.